UNIVERSITÀ
DI TORINO

Università degli Studi di Torino

Corso di Laurea Magistrale in Informatica

Karaoke e Inclusione: Un Esperimento Didattico per lo Sviluppo delle Competenze Linguistiche e Cognitive nelle Persone con Sindrome di Down

Relatore/Relatrice
Fabio Ciravegna

Candidato/a
Daloiso Pasquale
Matricola 335085

Anno Accademico 2025/2026

Indice delle figure

Indice generale

Introduzione

Il progresso tecnologico ha aperto nuove opportunità per migliorare la vita delle persone con disabilità, offrendo strumenti in grado di supportare l'autonomia, la comunicazione e la gestione delle attività quotidiane. L'obiettivo di questa tesi è la ricerca e lo sviluppo di un applicativo software rivolto a persone con disabilità, con particolare riferimento a quelle affette da sindrome di Down, allo scopo di rispondere alle loro esigenze specifiche e contribuire al miglioramento della qualità della vita.

L'applicativo si prefigge di facilitare la vita quotidiana delle persone con disabilità, incrementando la loro autonomia, migliorando le loro capacità di comunicazione e riducendo il carico di lavoro per i caregiver. Si prevede che l'utilizzo di una tecnologia accessibile e adattabile possa favorire l'inclusione sociale e l'autodeterminazione degli utenti.

La ricerca contribuirà allo sviluppo di strumenti tecnologici innovativi e inclusivi, con un forte impatto sociale. L'applicativo proposto mira a colmare le lacune esistenti nelle soluzioni tecnologiche per le persone con disabilità, promuovendo un accesso equo e facilitando una maggiore partecipazione degli individui disabili nella società.

L'utente

La sindrome di Down è una condizione genetica nella quale una persona possiede una copia supplementare di un cromosoma chiamato cromosoma 21. Questo cromosoma supplementare causa molti problemi di salute.

Il quadro clinico delle persone affette da Sindrome di Down presenta alcuni elementi ricorrenti, quali:

- deficit intellettivo di grado variabile
- sviluppo motorio ritardato
- problemi di salute
- problemi di comunicazione e liguaggio
- problematiche sociali e comportamentali

- invecchiamento precoce e declino cognitivo
- benessere emotivo (spesso depressione)

La scuola

La normativa italiana dal 2009 ha sostituito il termine di integrazione e ha introdotto il termine di inclusione nelle scuole italiane. Per integrazione si intende che la scuola deve accogliere l'alunno rimodellando il suo approccio didattico e valorizzando la diversità che diventa risorsa per il gruppo. Negli ultimi anni, le normative in fase di sviluppo pongono un accento crescente sull'autonomia scolastica, riconoscendo alle istituzioni scolastiche la possibilità di selezionare i propri programmi educativi in base alle specifiche necessità dei propri studenti. Tale orientamento implica che la programmazione educativa non si limiti alla strutturazione del curricolo obbligatorio, ma possa estendersi all'organizzazione di attività scolastiche integrative, destinate a gruppi di alunni, con l'obiettivo di realizzare interventi pedagogici mirati e individualizzati. Questi interventi, infatti, devono essere progettati in funzione delle peculiarità e delle esigenze di ciascun alunno, garantendo un percorso di apprendimento il più possibile personalizzato e rispondente alle diverse necessità formative e di sviluppo.
Con questi presupposti si evidenziano le esigenze di nuovi approcci didattici e nuovi strumenti di supporto per l'attività didattica. In particolare l'insegnante di sostegno dovrà prendersi carico di più responsabilità e non lasciare l'alunno con difficoltà a meri compiti marginali per non disturbare lo svolgimento del programma didattico standard ma dovrà anche proporre attività che coinvolgano gruppi di persone della classe o anche interclasse per riuscire a creare da un lato sviluppare nelle persone normodotate tatto ed empatia, soft skills apprezzate in qualunque ambiente lavorativo, e dall'altro stimolare nelle persone in difficoltà competenze di socializzazione, motorie e cognitive.
I principali problemi da risolvere in classi dove ci sono persone disabili sono:

- accettazione dell'alunno portatore di disabilità
- superamento delle resistenze psicologiche (stereotipi e pregiudizi)

Diventare insegnanti di sostegno dovrebbe essere dal mio punto di vista un percorso più complesso per preparare meglio lo stesso docente. Attualmente in Italia prevede semplicemente una specializzazione dopo la tua laurea specialistica da 60 cfu chiamata TFA. In Polonia, invece, il percorso per diventare insegnante di sostegno è strutturato e prevede una formazione specifica per fornire il supporto necessario agli studenti con difficoltà di apprendimento, disabilità intellettive, fisiche o disturbi dello sviluppo, come la sindrome di Down o l'autismo. Ecco come funziona il percorso per diventare insegnante di sostegno in Polonia:

1. Formazione accademica: 5 anni di studi universitari per conseguire una laurea in pedagogia speciale o educazione inclusiva. Durante questi studi si possono prendere diverse specializzazioni per andare a trattare le diverse problematiche della disabilità
2. Tirocinio: durante gli anni universitari si conseguono delle esperienze pratiche ad integrazione del percorso teorico
3. Esame finale: bisogna conseguire un'abilitazione all'insegnamento in cui ti mettono alla prova per identificare non solo i migliori insegnanti ma le persone con maggiore empatia e attitudine nello svolgere queste professioni delicate.
4. Formazione continua dopo gli studi. Chi tratta persone disabili non modifica solo la vita del ragazzo disabile ma dell'intera famiglia che lo supporta e quindi educatori e insegnanti che gestiscono persone disabili dovrebbero essere persone molto specializzate.

Per trattare persone con esigenze speciali è necessario conoscere principalmente 3 tipologie di competenze:

1. Pedagogiche, perché si è pur sempre educatori
2. Psicologiche, perché si trattano persone fragili
3. Antropologiche, per capire il contesto familiare e culturale in cui vive il discente

A supporto di queste 3 competenze principali ogni insegnante al giorno d'oggi dovrebbe avvalersi di un computer e di applicazioni che aiutino lo svolgimento del suo compito.
Da questo pensiero si potrebbe approfondire il rapporto tra scuola, computer e bambino e le sue capacità di apprendere in età moderna, ma questo non è argomento di ricerca. Invece importante per quello che tratterò e lo sviluppo del linguaggio che in persone normodotate si sviluppa in 3 fasi e in determinate età:

1. linguaggio autistico volto a soddisfare i bisogni essenziali dell'io che compare al secondo anno di vita (fame, sete, dolore)
2. linguaggio egocentrico incentrato sul proprio punto di vista
3. linguaggio sociale che compare attorno al settimo anno di vita in cui prendono visioni empatiche delle problematiche

La comunicazione nelle persone con sindrome di Down

Per persone disabili con sindrome di down in molti casi la comunicazione è ferma per lo più allo stadio pre-operatorio con anche difficoltà motorie nel formulare alcune parole complesse o particolari fonemi in quanto hanno anche caratteristiche fisiche che ostacolano la capacità comunicativa come una lingua più grande.

Queste difficoltà possono portare ad una bassa realizzazione di se stessi sfociando in episodi di impatto sociale negativo.

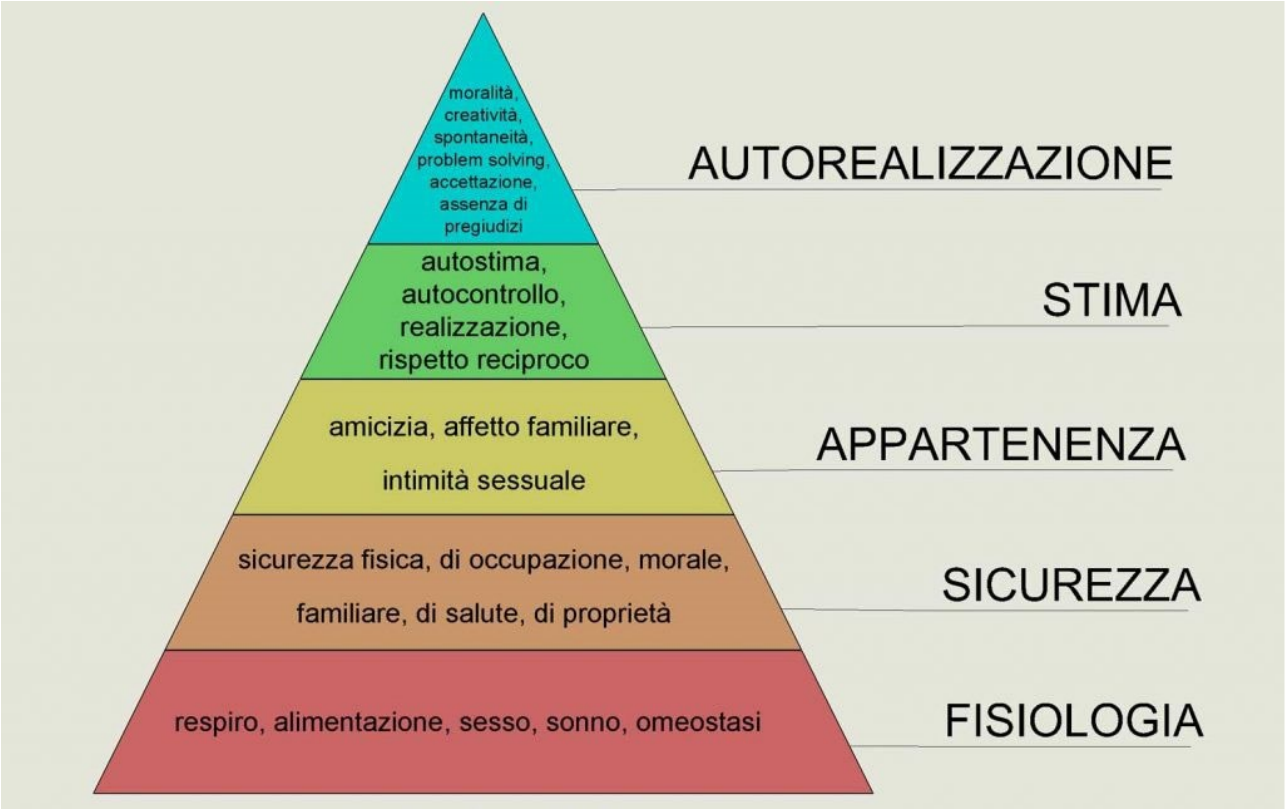


Figura 1: Piramide di Maslow

Nella nota piramide di Maslow una persona con sindrome di down ha difficiltà ad arrivare al secondo scalino. Essendo persone fragili vanno protette da situazioni che essi stessi causano.
La comunicazione verbale limitata fa si che la comunicazione non verbale sia molto pronunciata ma chi non è allenato a percepirla fa difficoltà nel comunicare rendendo la comunicazione in una sola direzione e aumentando la frustrazione della persona con disabilità.
La sindrome di Down è classificata come una disabilità intellettiva, caratterizzata da una compromissione cognitiva di vari livelli. Tuttavia, oltre agli aspetti cognitivi, questa condizione comporta anche disabilità di tipo motorio che coinvolgono sia le attività grosso-motorie, come il camminare e il mantenimento dell’equilibrio, sia le attività fino-motorie, che richiedono precisione e coordinazione, come ad esempio allacciarsi le scarpe. Tale compromissione motoria si manifesta con una ridotta tonicità muscolare (ipotonia) e una difficoltà nella coordinazione dei movimenti, fattori che influenzano negativamente l'autonomia individuale nelle attività quotidiane.

In passato, l'aspettativa di vita per le persone affette da sindrome di Down era significativamente ridotta. Tuttavia, grazie ai progressi della medicina e all'attenzione sanitaria, tale aspettativa si è notevolmente estesa. Nonostante questi miglioramenti, è stato osservato che, nelle persone adulte con sindrome di Down, l’insorgenza di demenza senile si manifesta in età relativamente precoce,

spesso già tra i 40 e i 50 anni. Questo fenomeno è frequentemente associato a una ridotta stimolazione cognitiva e ambientale, che contribuisce al declino delle funzioni cognitive in età adulta. Inoltre, la mancanza di una stimolazione educativa costante potrebbe favorire l'insorgenza di disturbi dello spettro autistico nelle persone con sindrome di Down. L'assenza di un adeguato supporto educativo e cognitivo può infatti compromettere ulteriormente lo sviluppo delle capacità relazionali e comunicative, aggravando il quadro clinico e comportamentale di questi individui. Una stimolazione continua e mirata riveste quindi un ruolo fondamentale nel prevenire o ridurre l'insorgenza di tali disturbi associati.

Per migliorare attività cerebrali e motorie, il gioco riveste un ruolo formativo determinante per lo sviluppo della sua personalità. Il gioco è considerato come il modo più naturale di costruire i propri modelli di conoscenza e di comportamento.
Attraverso l'attività ludica il bambino prende coscienza della realtà circostante, si sente protagonista dell'azione, afferma sé stesso e le sue esigenze arricchendo, così, la sua immaginazione. Tale attività, inoltre, avvia il bambino alla conoscenza di ciò che accade attorno a lui, e quindi sviluppa le sue capacità cognitive: è con il gioco che gli esseri umani imparano ad adattare le situazioni ai loro scopi, ne analizzano le catteristiche e ne stabiliscono le relazioni tra vari elementi della realtà. Vygotskij, psicologo e pedagogista russo, noto soprattutto per i suoi contributi alla psicologia dello sviluppo e all'educazione, riteneva che il gioco svolgesse un ruolo anticipatorio contribuendo così alla creazione di una "zona di sviluppo prossimale". Per quest'ultima si intende lo spazio esistente tra la concreta attitudine a risolvere un problema e il livello di "sviluppo potenziale" stabilito dalla medesima capacità di vivere le esperienze.
Un'altra problematica delle persone con sindrome di down è la capacità di attenzione. Sono gli utenti più distratti di qualsiasi applicazione si possa pensare di ideare. Un modo per rendere attrattiva un applicazione ideata per questa tipologia di utenti è renderla interattiva, semplice e ludica.
E' fondamentale inoltre analizzare il gioco in quanto strumento motivazionale. Un gioco è più motivante quando:

- presenta un'ambientazione che polarizza l'attenzione del soggetto
- ha elementi di sorpresa e scoperta che stimolano il giocatore a proseguire per soddisfare la curiosità
- è di difficoltà adeguata in quanto se troppo difficile spezzerebbe la sensazione piacevole di coinvolgimento e se troppo semplice annoierebbe
- non ha troppe spiegazioni o regole o capacità che l'utente ha difficoltà ad usare

Per quest’ultimo punto ci sono ricerche che evidenziano che persone con sindrome di down hanno difficoltà in particolari gesture e quindi l’usabilità di applicazioni per questi utenti è necessaria farla in maniera consona[2].
In particolare si nota che le persone con la sindrome di down hanno difficoltà a capire il significato delle icone o con l’uso del mouse. I testi lunghi provocano un affaticamento e quindi disincentivano il meccanismo del divertimento. L’audio in un video non deve essere troppo forte né troppo debole. Se troppo debole viene ignorato, mentre se troppo forte viene assimilato a stati confusionali della persona. Task cronometrati provocano stress e quindi peggiorano le performance dell’utente. Colori ben definiti senza sfumature di colore riescono ad attrarre l’attenzione più di colori misti: quindi un magenta è sconsigliabile in quanto potrebbe essere confuso con il rosa o l’arancione.
La memoria a breve termine è lunga all’incirca 25 parole e possono avere problemi linguistici di espressione e dura meno rispetto ad una persona con capacità tipiche. La memoria a lungo termine nelle persone con sindrome di Down è un punto di forza, poiché tende ad essere particolarmente duratura. Queste persone, infatti, tendono a riflettere in modo persistente su determinati pensieri, il che contribuisce a rendere indelebili i ricordi. Questo processo può essere utile per mantenere a lungo le informazioni, anche se, in alcuni casi, può portare a focalizzarsi su determinati ricordi in modo ossessivo. Bisogna specificare comunque che nella memoria a lungo termine la memoria esplicita, cioè quella consapevole e intenzionale (richiamabile deliberatamente), risulta leggermente compromessa.
Inoltre queste parole vengono ricordate meglio nella memoria a breve termine se hanno 2 caratteristiche:

- sono diverse tra di loro. Quindi vino, lino e pino provocano confusione nel distinguerle e quindi mettere nella stessa frase parole simili aumenta la probabilità che non se le ricordino
- sono difficili da articolare. Un dittongo complesso o una parola particolarmente lunga può generare stress nell'utente, portandolo ad abbreviare o alterare la parola.

Rondal, J.A nella sua ricerca sullo sviluppo cognitivo delle persone con sindrome di Down nel 2004, come altri in precedenza conferma che essi non compiono la reiterazione articolatoria durante gli stimoli verbali: tale mancanza accelera il decadimento delle traccie di memoria dal magazzino fonologico. Nonostante questo il peggioramento evidente nei compiti in cui bisogna ripetere una sequenza all'indietro fa pensare che anche il sistema che controlla l'organizzazione e la gestione delle informazioni (sistema esecutivo) abbia delle responsabilità nella compromissione della gestione della memoria.
Le persone con sindrome di Down sembrano quindi avere più difficoltà ad apprendere mediante modalità esplicite, mentre appaiono relativamente più accessibili apprendimenti impliciti, cioè basati sull’esperienza e sul fare. Essi possono essere impiegate in vari ambiti lavorativi, come il

giardinaggio o la ristorazione, ad esempio presso McDonald's, e in numerosi casi dimostrano di essere più produttive rispetto ai loro coetanei senza disabilità.
Nel linguaggio presentano un deficit marcato nelle competenze liguistiche rispetto alle altre abilità motorie, cognitive e sociali se paragonati a bambini con ritardi mentali di diversa eziologia o con sviluppo tipico. Distinguendo tra linguaggio prodotto e linguaggio compreso, si osserva che, soprattutto nei primi 15-20 anni di vita e anche successivamente, le persone con sindrome di Down presentano maggiori difficoltà nell'esprimersi rispetto alla comprensione del linguaggio.
Le difficoltà comunicative possono avere un impatto profondo sul benessere psicologico e sociale degli individui, conducendo a fenomeni di isolamento e frustrazione che, in taluni casi, possono sfociare in comportamenti antisociali. Questo rischio è particolarmente elevato in presenza di sentimenti di depressione, i quali possono essere acuiti dalla limitata capacità di interazione con l'ambiente sociale circostante.

Lera Boroditsky[11], nel suo intervento TED, sostiene che il linguaggio non rappresenta un semplice strumento di comunicazione, ma una vera e propria guida per il pensiero e, di conseguenza, per le azioni. Secondo la sua prospettiva, le strutture linguistiche non solo modellano il modo in cui gli individui percepiscono la realtà, ma influenzano anche il modo in cui si relazionano con il mondo. Le differenze linguistiche, ad esempio nelle concezioni temporali, spaziali o emotive, possono indurre modi di pensare e di agire differenti tra i parlanti di lingue diverse, con un impatto diretto sulle loro esperienze quotidiane.

Tale connessione tra linguaggio e pensiero diviene cruciale quando si considerano le implicazioni delle difficoltà linguistiche. L'incapacità di comunicare efficacemente non solo ostacola la capacità dell'individuo di comprendere e farsi comprendere, ma può anche compromettere la sua partecipazione sociale, contribuendo all'insorgere di stati di alienazione e disagio psicologico fino ad arrivare alla depressione.

Le persone adulte con sindrome di Down mostrano un'insorgenza di demenza senile e Alzheimer più precoce rispetto alle persone con sviluppo tipico. Considerando che ogni forma di demenza presenta un decorso clinico variabile, è possibile che questa variabilità sia ancora più accentuata nelle persone con sindrome di Down, soprattutto se adeguatamente stimolate.
L'uso del computer o di dispositivi tecnologici come il telefono cellulare può rivelarsi particolarmente vantaggioso per le persone con sindrome di Down, poiché facilita l'apprendimento di nuove abilità e competenze. Grazie a una buona memoria implicita, queste persone possono acquisire rapidamente le piccole funzionalità tecnologiche, richiedendo un impegno minimo. Inoltre, le applicazioni e i programmi didattici disponibili possono essere progettati per adattarsi alle loro esigenze specifiche, migliorando ulteriormente l'efficacia dell'apprendimento.

Abilitazione e preservazione

Nelle persone con sindrome di Down, il processo di riabilitazione si trasforma in un processo di abilitazione, poiché le competenze devono essere sviluppate e potenziate. Questo approccio è fondamentale affinché, in età avanzata, tali competenze possano essere preservate e mantenute.

È fondamentale sviluppare un approccio e soluzioni personalizzate per ogni problematica e per ciascun utente. È necessario disporre di molteplici strumenti per affrontare le diverse difficoltà. La valutazione continua da parte di un terapista consente di selezionare gli strumenti più appropriati da utilizzare. Le strategie adottate non devono necessariamente seguire un programma prestabilito; è essenziale avere una varietà di strumenti a disposizione per facilitare l'interazione con la persona in difficoltà.

È fondamentale fornire un rinforzo visivo e gestuale in qualsiasi terapia mirata a stimolare la comunicazione, poiché studi hanno dimostrato che la memoria visiva facilita la memorizzazione implicita nei soggetti.

Un punto di forza da valorizzare è la perseveranza. Quando viene individuata un'attività che suscita il loro interesse, le persone con sindrome di Down tendono a ripetere l'esercizio in modo intenso e costante, mostrando una notevole resistenza e assenza di perdita di interesse.

Data l'ampia varietà di capacità presenti tra gli utenti o dello stesso utente in diverse età, è necessario distinguere l'applicazione in diversi livelli di difficoltà e adattare progressivamente tali difficoltà, proprio come avviene nei giochi.

Le persone con sindrome di Down, in quanto individui unici con proprie personalità, presentano caratteristiche e interessi diversi. È un errore comune considerare che tutti gli individui con sindrome di Down siano necessariamente sensibili o sempre felici; al contrario, esistono anche persone che possono adottare comportamenti bullistici, altri che magari nutrono un'elevata autostima tanto da considerarsi magari bellissimi oppure altri che hanno un’autostima bassa. Quindi bisogna provare a creare un modo per rafforzare le loro debolezze o smorzare comportamenti antisociali durante la terapia.

In alcune fasi della vita, è fondamentale fornire anche un'educazione sessuale adeguata alle persone con sindrome di Down, poiché questa non deve essere considerata un'opzione, ma un diritto essenziale. L'approccio all'educazione sessuale deve essere personalizzato in base alle capacità individuali e alle eventuali problematiche riscontrate. Inoltre, in alcuni casi, è utile attuare interventi di gruppo, poiché il contesto sociale può influenzare negativamente le dinamiche tra pari e contribuire a malintesi o comportamenti inappropriati.

Fornire supporti per attività quotidiane semplici, come la selezione del campanello da suonare per comunicare con il vicino di casa, e per l'educazione a evitare comportamenti pericolosi, come accendere un fornello o aprire la porta di casa a sconosciuti è un altro campo in cui la terapia può aiutare a migliorare la vita di queste persone. Attività che possono sembrare banali per una persona senza disabilità possono rivelarsi non scontate per individui con sindrome di Down.

Alcuni studi hanno effettivamente esplorato l'utilizzo di approcci farmacologici per affrontare i problemi cognitivi associati alla sindrome di Down, sperimentando su modelli murini farmaci volti a modificare la struttura e la funzionalità del cervello. Questi farmaci mirano, in particolare, a migliorare la plasticità sinaptica, la memoria e l'apprendimento negli animali, tentando di contrastare i deficit cognitivi legati alla condizione. Ad esempio, alcuni studi hanno esaminato l'effetto di farmaci che modulano il sistema dell'acido gamma-aminobutirrico (GABA), cercando di ridurre l'inibizione neurale e migliorare la comunicazione tra le cellule cerebrali [13]. Tuttavia queste sperimentazioni sono dal mio punto di vista, per quanto affascinanti, possono sembrare piuttosto ottimistiche e speculative. Anche qualora i farmaci riuscissero a produrre miglioramenti cognitivi in modelli animali, le differenze tra il cervello dei topi e quello umano rendono complesso prevedere risultati simili nelle persone. Gli interventi farmacologici potrebbero, nel migliore dei casi, fornire un aiuto parziale, ma è improbabile che possano "curare" completamente una condizione complessa e multisistemica come la sindrome di Down.

In sintesi, mentre la ricerca farmacologica offre spunti interessanti, il vero potenziale di intervento sembra risiedere in un approccio più olistico, che comprenda il supporto educativo, sociale e terapeutico per affrontare le sfide legate a questa condizione.

Con ciò non intendo affermare che il supporto educativo rappresenti la soluzione a tutti i problemi, poiché molteplici fattori contribuiscono alla complessità della situazione. Molti studi confermano questa complessità tra cui anche la **World Health Organization.** International Classification of Functioning, Disability and Health, Geneva, World Health Organization; 2001. Alcuni articoli [17] che parlano di questa condizione hanno il riferimento a questo schema:

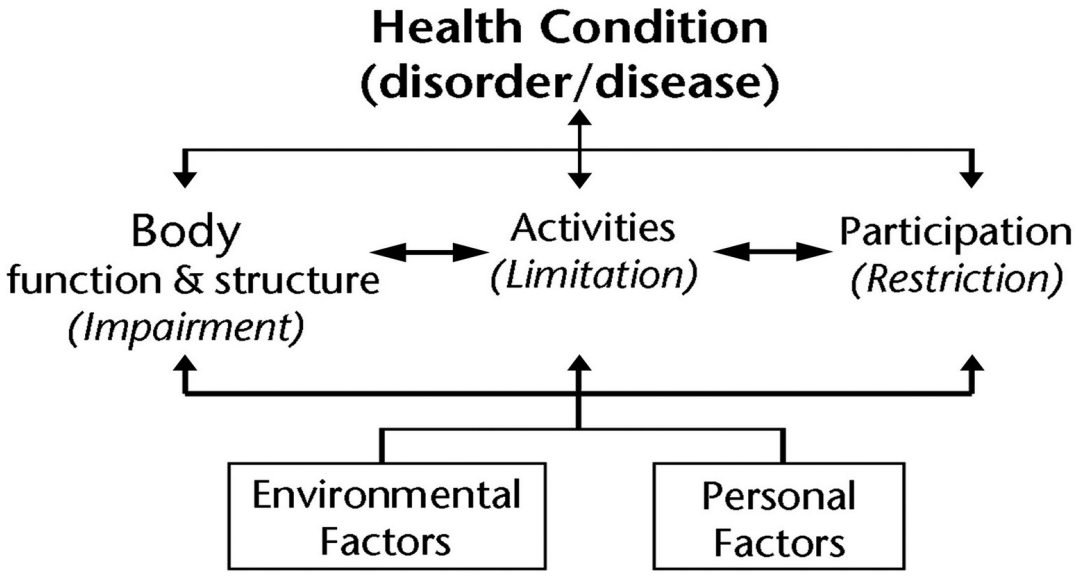


Figura 2: Complessità della funzione cognitiva

Nel **modello ICF** (Classificazione Internazionale del Funzionamento, della Disabilità e della Salute), la disabilità e il funzionamento sono visti come il risultato dell'interazione tra condizioni di salute (malattie, disturbi e lesioni) e fattori contestuali. I **fattori contestuali** includono sia i **fattori ambientali** che i **fattori personali**.

- I **fattori ambientali** sono esterni e comprendono atteggiamenti sociali, cultura, geografia e caratteristiche architettoniche.
- I **fattori personali** sono interni e includono sesso, età, caratteristiche della personalità, background sociale, istruzione, esperienze di vita, attività professionali e non professionali, e qualsiasi altro fattore che possa influenzare come una persona vive la propria disabilità.

Gli interventi terapeutici in età adulta

Gli interventi terapeutici per gli adulti con sindrome di Down sono orientati a un approccio multidisciplinare e personalizzato. Vanno oltre il trattamento delle condizioni fisiche, includendo il supporto cognitivo, emotivo e sociale, con l’obiettivo di preservare l’autonomia e promuovere il benessere complessivo. Un approccio continuo e integrato tra professionisti della salute, caregiver e reti di supporto è essenziale per rispondere alle sfide uniche di questa fase della vita.

Le attività proposte possono includere, a titolo esemplificativo, le seguenti:

- Nuoto, volto a favorire lo sviluppo della muscolatura e migliorare le capacità motorie globali;

- Cartonaggio, finalizzato allo sviluppo della motricità fine e al potenziamento delle abilità manuali;
- Passeggiate di gruppo o scuola calcio, con l’obiettivo di promuovere la socializzazione tra persone con disabilità e favorire l’inclusione sociale;
- Laboratorio di ceramica o pelletteria, per migliorare la manualità e affinare le competenze tecniche attraverso l'uso di materiali specifici;
- Creazione di magliette serigrafate, attività che stimola la creatività e favorisce l’espressione personale attraverso l’arte;
- Cucina, che può rappresentare un’opportunità di apprendimento professionale, nonché un mezzo per rafforzare l’autostima e la fiducia nelle proprie capacità.

Tali attività vengono selezionate in base alle abilità individuali della persona, alle sue attitudini personali e agli obiettivi di miglioramento specifici concordati, tenendo conto delle esigenze e delle potenzialità di ciascun partecipante.

La ricerca

Con questi presupposti ho fatto una ricerca per cercare applicazioni già sviluppate e trovare un’applicazione utile e non ancora sviluppata. Ho trovato applicazioni [3][4][5] per imparare la matematica, applicazioni per migliorare l’equilibrio in quanto principale fonte di cadute soprattutto quando hanno 6/8 anni, applicazioni per leggere/scrivere e coordinarsi. Ho addirittura trovato un’applicazione per esercitare la lingua in quanto le persone con sindrome di down hanno la lingua più grande del normale e fanno difficoltà a gestirla per parlare.
Quello che non sono riuscito a trovare sono applicazioni che aiutassero nella comunicazione intesa non come atto del parlare ma intesa come scambio di pensieri tra persone e nella socializzazione.
I benefici che ci sarebbero nello stimolare queste capacità sarebbero enormi, in quanto attraverso la comunicazione la persona con esigenze particolari riuscirebbe a evitare atteggiamenti antisociali migliorando anche la sua qualità di vita.
Effetto secondario non di poco conto la socializzazione riuscirebbe a stimolare anche altre dinamiche come la peer education in cui sono gli stessi coetanei a creare un naturale passaggio di conoscenze, emozioni ed esperienze, mettendo così in moto un processo di comunicazione globale che diviene una vera e propria occasione di arricchimento e di scambio per il singolo adolescente.
Paul Watzlawick, psicologo e filosofo austriaco specializzato in comunicazione, ha dichiarato che l’essere umano non riesce a non comunicare. Le modalità con cui si comunica dipende dalla capacità dell’individuo e tutta la comunicazione influenza il comportamento umano. Per il

sociologo Zygmunt Bauman il fallimento di una relazione è quasi sempre un fallimento di comunicazione.
I disturbi del linguaggio si possono distinguere in diverse tipologie:

1. fonetico-fonologico, una difficoltà nella produzione dei suoni dell'eloquio
2. disturbo della fluenza (balbuzie)
3. disturbo della comunicazione sociale come scambiarsi informazioni o convenevoli

Per una persona con sindrome di Down, i principali problemi fonetici vengono generalmente affrontati durante l'età scolare con l'intervento di logopedisti. Tuttavia, sarebbe necessaria una presa in carico logopedica continuativa lungo tutto l'arco della vita, poiché le abilità non costantemente esercitate tendono a deteriorarsi nel tempo in individui con questa condizione. Oltre ai problemi fonetici, un'altra difficoltà rilevante è rappresentata dalle problematiche legate alla comunicazione sociale, che spesso non ricevono l'attenzione adeguata all'interno del sistema scolastico italiano, risultando così trascurate nel processo educativo complessivo.

La musica e l'educazione inclusiva

Nel mondo ci sono scuole [6] che basano la loro educazione a persone disabili sulla musica in quanto la musica è uno strumento di inclusione di per sé. La musica è un linguaggio universale che trascende le barriere. Per le persone con disabilità, può rappresentare uno spazio in cui si superano i limiti fisici o mentali. I programmi educativi e i laboratori di musicoterapia sono diffusi in molti contesti terapeutici per promuovere lo sviluppo cognitivo, emotivo e motorio. La **musicoterapia** è particolarmente efficace con persone affette da disturbi dello spettro autistico, difficoltà di apprendimento, disabilità motorie e altre condizioni neurologiche. La musica aiuta a migliorare l'autoespressione e la comunicazione, favorendo l'inclusione sociale.

Ci sono persino libri [7] che ne approfondiscono la loro relazione. In particolare la musicoterapia è una pratica diffusa per migliorare la qualità della vita delle persone con disabilità. Grazie alla musica, si riesce a stimolare aree del cervello legate alla memoria, alla motricità e alle emozioni. È utile per:

- Riabilitazione motoria
- sviluppo cognitivo
- supporto emotivo

Attualmente il karaoke viene usato per il trattamento di casi di demenza senile o altri problemi mentali [9][10]. I benefici derivanti dall'attività musicale includono:

- **Partecipazione sociale**: Grazie all’utilizzo della musica attiva, sia tramite la produzione musicale che l'ascolto, la musica diventa uno strumento terapeutico che favorisce attività rieducative, orientate alla presa di coscienza o al lavoro ricostruttivo.
- **Esercizio fisico**: La combinazione di musica e movimento stimola l'attività fisica, contribuendo al miglioramento delle capacità motorie e favorendo il benessere fisico complessivo.
- **Rilassamento**: L’ascolto di determinati tipi di musica, scelti in base alle necessità terapeutiche, può favorire uno stato di rilassamento e ridurre i livelli di stress.
- **Incremento dell’immaginazione**: L’utilizzo della musica può stimolare l’immaginazione, riducendo la tensione e favorendo la focalizzazione su pensieri e sentimenti positivi, contribuendo così al benessere psicologico.

La **musicoterapia** è utilizzata anche per migliorare la parte muscolare[12]. Oltre ai benefici cognitivi ed emotivi, la musicoterapia può avere un impatto positivo sullo sviluppo motorio e sulla coordinazione fisica.

L’idea

L’idea proposta prevede lo sviluppo di un’applicazione dedicata al karaoke, specificamente progettata per rispondere alle esigenze delle persone con sindrome di Down o altre disabilità cognitive. L'applicazione, basata su un'interfaccia semplice e accessibile, consentirebbe agli utenti di interagire con la musica in modo personalizzato e inclusivo.

La personalizzazione potrebbe essere guidata attraverso un form compilato dai genitori prima di iniziare l’utilizzo effettivo in cui poter richiedere età, interessi e eventuali problematiche da risolvere. Questo form utilizzato in fase di registrazione può essere modificato anche in un secondo momento.

In termini di funzionalità, l'app potrebbe generare brani musicali con testi situazionali creati dall’intelligenza artificiale (AI) in base al contesto e alle esigenze dell’utente, o consentire il caricamento di brani scelti dall'utente stesso, personalizzando l’esperienza in base ai gusti e alle preferenze individuali. Questa flessibilità permetterebbe di creare un'esperienza musicale più coinvolgente e significativa per ogni utilizzatore, rispondendo alla varietà delle capacità cognitive e delle preferenze personali.

Un elemento innovativo dell’applicazione è l’utilizzo di **tecnologie di riconoscimento facciale** attraverso la telecamera del dispositivo, che consentirebbe di monitorare le reazioni emotive

dell’utente durante l’attività di karaoke. Il sistema sarebbe in grado di rilevare eventuali segnali di frustrazione o disagio, intervenendo in tempo reale per correggere l’esperienza in modo positivo e adattivo. In presenza di segnali di frustrazione, l’app potrebbe fornire suggerimenti utili o rallentare il ritmo del brano, oppure, in alternativa, attivare messaggi motivazionali e segnali di incoraggiamento volti a ridurre lo stress e a mantenere alto il livello di coinvolgimento.

Questa dinamica interattiva mira a creare un ambiente musicale privo di pressioni, favorendo il benessere emotivo e facilitando l’apprendimento, la socializzazione e l’espressione personale. L'approccio mirato dell'app, combinando la creazione musicale adattiva con strumenti tecnologici di monitoraggio emotivo, garantirebbe un’esperienza utente empatica e personalizzata, prevenendo l’insorgenza di stress e frustrazione e valorizzando l’attività ludico-educativa del karaoke come strumento di supporto terapeutico.

Dinamica interazionale proposta

La dinamica interazionale si articola in due fasi principali: una prima fase di apprendimento e memorizzazione delle parole, seguita da una seconda fase in cui le parole apprese vengono utilizzate all'interno di canzoni generate dall'intelligenza artificiale.

Prima fase: associazione e memorizzazione delle parole

Nella prima fase, l'intelligenza artificiale propone una serie di parole o frasi selezionate, associate a supporti visivi o sonori al fine di facilitare il processo di apprendimento. Gli utenti sono invitati a ripetere le parole proposte, con l'obiettivo di migliorare la propria capacità di riconoscimento e pronuncia. L'IA monitora e corregge attivamente eventuali errori di pronuncia, offrendo un feedback immediato per favorire il miglioramento continuo e la memorizzazione.

Questa fase sfrutta principi pedagogici legati all'apprendimento iterativo e associativo: il ripetersi delle associazioni tra stimolo visivo/sonoro e parola contribuisce al consolidamento delle informazioni nel sistema cognitivo degli utenti. L'intelligenza artificiale svolge un ruolo chiave nella personalizzazione del percorso, adattando la difficoltà e la tipologia delle parole in base alle prestazioni dell'utente.

Seconda fase: esecuzione di canzoni generate dall'IA

Nella seconda fase, le parole apprese vengono integrate in canzoni generate dall'intelligenza artificiale, offrendo un contesto dinamico e creativo per il loro utilizzo. Le canzoni, basate su melodie semplici e ripetitive, sono create in modo tale da incorporare le parole precedentemente definite, permettendo agli utenti di rinforzare ulteriormente l'apprendimento attraverso il canto.

La scelta del canto come strumento pedagogico si basa sulla sua comprovata efficacia nell'attivazione della memoria a lungo termine e nella promozione di una maggiore partecipazione attiva da parte dell'utente. Cantare le parole apprese all'interno di un contesto musicale stimola la memorizzazione e la fluidità linguistica, migliorando al contempo la pronuncia in modo naturale e divertente.

In sintesi, questa dinamica interazionale propone un ciclo di apprendimento integrato, dove la fase iniziale di memorizzazione viene consolidata attraverso un'attività ludica e partecipativa, rendendo l'apprendimento delle parole più coinvolgente e efficace.

Il confronto con le app esistenti

Smule: App di karaoke molto popolare che consente di cantare in gruppo o da soli. Anche se non è specificamente progettata per persone con disabilità, la sua interfaccia semplice e la vasta libreria di canzoni la rendono adatta per alcuni utenti.
Karaoke – Sing Unlimited Songs: Un'app che permette di cantare senza limiti di tempo o di canzoni, con molte opzioni di personalizzazione e che potrebbe essere adattata per un utilizzo più accessibile.
Singa: Questa app ha un'interfaccia chiara e semplice che può essere facile da navigare anche per persone con disabilità cognitive lievi.
In ogni caso non esistono app specificatamente progettate per persone con sindrome di down.

Sfide tecnologiche

In un progetto volto allo sviluppo di un applicativo di supporto per persone con disabilità, come ad esempio quelle affette da sindrome di Down, emergono diverse sfide tecnologiche che richiedono un'analisi approfondita e soluzioni specifiche. Tali sfide derivano dalle particolari esigenze degli utenti finali. Di seguito vengono elencate le principali sfide tecnologiche identificate, che saranno approfondite in termini di possibili soluzioni:

- creare un testo di un brano a partire usando termini ricercati (più altri nuovi che non sono stati studiati)
- creare un brano in base al testo della canzone e a caratteristiche di difficoltà (più è veloce il brano meno sarà facile per l’utente cantarci insieme)
- cattura dell’immagine dell’utente
- interpretazione dell’umore dell’utente

- Riproduzione di testo e musica
- ascolto e correzione del testo ripetuto dall’utente
- ricerca di parole con caratteristiche in base all’utente
- ricerca di immagini legate alle parole
- uso di un tool multipiattaforma così da creare un applicativo web e mobile, partendo l’indagine da kotlin multiplatform
- riconoscimento facciale delle emozioni
- riconoscimento del testo all’interno della canzone

Il presente elenco appare sufficientemente completo riguardo alle problematiche da affrontare per il buon esito del progetto. Tra queste, la creazione di un brano rappresentava la questione che destava maggiori preoccupazioni, poiché la sua fattibilità non era ancora del tutto chiara. Pertanto, ho inizialmente avviato la sperimentazione di questa fase, per poi affrontare le altre problematiche mediante l'uso di strumenti in grado di analizzare gli aspetti economici e funzionali.

Creazione di brani

La fattibilità del progetto è supportata dalla disponibilità di strumenti già esistenti per la generazione di brani musicali basati su parametri predefiniti. Ad esempio, il sito Suno permette di creare canzoni in base a specifiche indicazioni. Inserendo la richiesta di "una canzone lenta con le parole: rosso, giallo, blu, i colori Power Rangers", il sistema genera un brano dal ritmo lento e di facile esecuzione. Tuttavia, questo servizio non fornisce API per la creazione automatizzata di musica.

Al contrario, i risultati ottenuti utilizzando Melobytes si sono rivelati insoddisfacenti in termini di qualità musicale. Un altro strumento disponibile è AIVA, il cui utilizzo a livello gratuito è limitato a tre download al mese, rendendo necessario sottoscrivere un abbonamento a partire da 11 euro al mese per un uso più esteso. Questa soluzione, pur valida, non soddisfa l'esigenza di un servizio con una quota freemium più generosa e un modello di pagamento a consumo.

Un altro esempio è Boomy, che nella versione gratuita non consente il download dei brani creati, analogamente a quanto accade con AIVA. Il servizio MuseNet di OpenAI, invece, non offre API basate su testo e genera musica su diverse ottave, lasciando all'utente la scelta tra i risultati proposti.

Infine, esistono strumenti open-source come Magenta, compatibile anche con Android, e Jukebox, entrambi richiedono installazione locale per il loro utilizzo. Tra queste soluzioni, Magenta appare

come l'opzione più promettente, motivo per cui verrà esplorata ulteriormente (https://magenta.withgoogle.com/get-started).

AIVA

Non supporta la creazione di musica a partire dal testo.

Magenta

Magenta è un tool per generare musica. Successivamente bisogna combinare il testo con la musica e poi sintetizzare con altri strumenti il testo e combinare insieme il testo e la musica. E' l'unico approccio disponibile concreto, ma richiede un livello di competenza specifico molto avanzato.

Correzione del testo ripetuto dall'utente

Per fare il confronto e la correzione dei file audio, si può scegliere strumenti come **Librosa** per analisi di basso livello (forma d'onda, spettrogramma) e librerie come **Pydub** o **noisereduce** per correggere i problemi audio più comuni. Questo è un argomento molto vasto per poter proseguire in questo ambito se si vuole ottenere un primo risultato in tempi rapidi.

Riconoscimento dell'umore

Esistono strumenti gratuiti come dlib (http://dlib.net/) oppure servizi a pagamento online per riconoscere come Amazon Rekognition per l'umore della persona. In questo caso è necessario usare strumenti a pagamento per utilizzare questa funzionalità in maniera veloce ed in futuro si può creare una propria implementazione per evitare i costi.

Ricerca di testo legato ad un soggetto e ricerca di brano legato a particolari parole o immagini legate alle parole

Esistono delle api a pagamento https://platform.openai.com/ per poter fare questo. I costi sono elevati per la creazione di immagini.

Funzionalità	Modello	Costo
Ricerche finalizzate a testo	gpt-3.5-turbo	$3.000 / 1M input tokens $6.000 / 1M output tokens
Ricerca di brano legato a particolari parole	text-davinci-003	$12.000 / 1M input tokens $12.000 / 1M output tokens
Creazione di immagini	dall-e	$0.016 / image

Una parola o un segno di punteggiatura produce dagli 1 ai 2 token in media.

Per gestire la creatività massima della ricerca bisogna specificare temperature=2.

Per limitare i costi e quindi i token usati come risposta del messaggio si può usare l'attributo json max_tokens.

Multiplatform

Kotlin multiplatform è uno strumento utile per condividere la business logic dell'applicazione lasciando aperta la scelta dei diversi front-end disponibili. Questa scelta presuppone di utilizzare kotlin e poi python per lo sviluppo delle parti di backend.

Flask e Django sono framework web in Python. Possono essere utilizzati per creare applicazioni web che possono funzionare anche come applicazioni mobili (PWA).

Flask è usato quando hai bisogno di un alto grado di libertà nella scelta delle librerie che devono essere scelte di volta in volta e va bene se sei esperto python.

Django è utile quando i progetti sono più complessi e richiedono una forte struttura con molte funzionalità come autenticazione web pronta all'uso e orm integrato. Inoltre essendo un framework di back-end lascia libera la possibilità di sviluppare il front-end con tecnologie web come react, vue o quasar, il quale quest'ultimo prevede nativamente dell'integrazione con capacitor https://quasar.dev/quasar-cli-vite/developing-capacitor-apps/preparation per eventuali sviluppi. Capacitor offre accesso alle funzionalità native del dispositivo attraverso un sistema di plugin. Quindi se il dispositivo è un telefono mobile si potrà utilizzare il plug-in che interagirà con la fotocamera.

Un altro framework di front-end degno di essere preso in considerazione è react con l'integrazione di componenti come Chakra UI per costruire interfacce accessibili seguendo le linee guida ARIA. Inoltre si possono integrare altre librerie per costruire html accessibile. Un esempio popolare è **React Accessibility** (o **react-a11y**), essendo una libreria che aiuta gli sviluppatori a identificare e correggere problemi di accessibilità nelle applicazioni React creando regole che devono poi essere rispettate nel codice.

Spikes

Ho dovuto installare un pacchetto del sistema operativo che rende possibile l'utilizzo delle librerie python.

FFmpeg (utilizzato anche da pydub) è un potente e versatile software open-source utilizzato per registrare, convertire, elaborare e trasmettere file audio e video. È uno strumento molto popolare nel mondo della produzione multimediale, grazie alla sua capacità di gestire una vasta gamma di formati e codec per audio, video e immagini. E' uno strumento essenziale per chiunque lavori con file audio e video, grazie alla sua capacità di gestire conversioni, compressione, editing, e streaming in modo efficiente e versatile.

FluidSynth è un sintetizzatore software open-source che consente di riprodurre suoni musicali utilizzando SoundFont, un formato di file che contiene campioni audio di strumenti musicali. FluidSynth è uno strumento potente e versatile per la sintesi audio e la riproduzione musicale, ideale per musicisti, produttori e sviluppatori che desiderano lavorare con strumenti virtuali e SoundFont (trasformando anche midi in wav file).

Possibile architettura scelta da chagpt

Ho creato un applicazione che concatena la voce sintetizzata di parole scritte a video e un file mp3.

Librerie:

- **Pydub**: Per la manipolazione audio.
- **Flask**: Per il backend dell'app web.
- **Web Audio API**: Per l'elaborazione audio nel browser.

Librerie:

- Flask per l'interfaccia web
- pydub per concatenare la musica e il testo
- gtts per sintetizzare il testo

Uso di librerie python

IntonazioneVoce.py

Ho creato uno script python che varia l'intonazione voce sintetizzandola

Librerie:

- numpy per la modifica dell'intonazione
- gtts per la sintesi vocale
- pydub per l'export in mp3

NoteConVoce.py

Ho creato uno script che suona note diverse a diversa lunghezza facendo l’overlay di una voce sintetizzata

Librerie:

- gtts per la sintesi vocale
- pydub per l’overlay della canzone e della voce e per la generazione delle note (pacchetto generators)
- random per la scelta di note casuali

ManipolazioneOnde.py

Ho cercato di manipolare le onde sonore con scarsi risultati utilizzando numpy come per la voce

Librerie:

- pydub per l’overlay della canzone e per alzare e abbassare il volume
- gtts per la sintesi vocale
- numpy per la modifica dell’onda sonora

CreazioneMusicaConSoundFont.py

Un **SoundFont** (.sf2) è un file che contiene una raccolta di campioni audio (suoni registrati) utilizzati per riprodurre strumenti musicali in formato digitale. Si possono scaricare font gratuiti ad esempio qui:

https://www.zanderjaz.com/downloads/soundfonts/guitars/

Ho creato delle note digitali con music21 generato un flusso musicale e salvato in un file mid.

Un file **MIDI** non contiene suoni veri e propri. È un formato di dati che memorizza informazioni su come la musica dovrebbe essere eseguita. In altre parole, è uno "spartito digitale" che indica le note da suonare, i tempi, le dinamiche, e i cambiamenti di strumenti. Esiste poi una libreria python chiamata Mido che è utile ad accedere a eventi MIDI come note, cambi di strumenti, dinamiche per poi manipolarne il risultato finale come un vero direttore d’orchestra.

Per traformarlo in onde sonore e quindi in un file wav ho usato fluidsinth.

Alla fine ho esportato come sempre il file in mp3 in quanto è un formato più compatto del file.

Librerie:

- music21 per la creazione midi
- pydub per l’export in mp3 e la sovrapposizione del file

Vocoder.py

Ho implementato uno script che applica l’autotune della voce rispetto ad una canzone. Gli step eseguiti sono i seguenti:

- tagliato il segnale più lungo
- tagliato in frame sia la voce che la musica
- applicato ad ogni frame la trasformata di Fourier STFT. La Short-Time Fourier Transform (STFT) è una tecnica di analisi dei segnali che permette di studiare le variazioni nel contenuto in frequenza di un segnale nel tempo. È particolarmente utile per l'analisi di segnali non stazionari, come la musica o la voce, dove le frequenze possono cambiare nel tempo.
- determinato come varia l'ampiezza (o il livello) del segnale nel tempo senza considerare la variazione di frequenza
- modulato l'ampiezza del segnale portante (musica) utilizzando l'inviluppo di ampiezza calcolato dal segnale modulante (voce).
- ricostruito il segnale audio applicando l’inverso della STFT
- applicato un filtro passa alto per eliminare un po’ di rumori della voce modulata
- applicato la musica alla voce cantata
- esportata la canzone

Il risultato è il massimo che si può ottenere in maniera digitale con una voce sintetizzata senza note. In alternativa bisognerebbe avere dei font anche per la voce.

Per ottimizzare il risultato finale del prodotto, è necessario intervenire sulla musica dopo la sua creazione iniziale, ma prima di procedere con l'integrazione della voce e del canto mediante l'uso del vocoder. In questa fase, risulta fondamentale adattare la musica al testo, in modo che le due componenti siano perfettamente sincronizzate. Un possibile approccio algoritmico potrebbe prevedere la sillabazione del testo, seguita dalla selezione delle note in base alla durata congrua di ciascuna sillaba. In questo modo, si garantirebbe una maggiore coerenza tra la melodia e la struttura linguistica del testo, ottimizzando l'esperienza sonora e favorendo una migliore comprensione e memorizzazione da parte dell'utente.

Magenta

Tra le librerie dipendenti si notano tensorflow per il machine learning e l'intelligenza artificiale, *Mido* prima menzionata e *Imageio* per leggere e scrivere immagini in vari formati.

Il manuale di installazione del prodotto sono troppo scarse ed è necessario installare pacchetti come llvm versione 8 che essendo una versione piuttosto vecchia ed essendo su windows senza un'installazione di linux nativa ma usando solo wsl il compilatore c++ o il compilatore python non è definito correttamente.

Anche provando ad installare un ambiente virtuale

```
python3 -m venv myenv

source myenv/bin/activate  # Su Linux/Mac
```

non sono riuscito a far funzionare magenta in un primo momento su windows.

Ci sono diversi esempi di web app sviluppate con magenta a questo sito: https://magenta.withgoogle.com/demos/web/

Dopo averinstallato linux, ho provato con diverse versioni di python a installare magenta. Ho creato un environment pip e installato **ffmpeg** e **FluidSynth**. Dopo ho lanciato i seguenti comandi:

```
sudo apt-get install -qq fluid-soundfont-gm build-essential libasound2-dev libjack-devC
```

per installare librerie e compilatori necessari alla compilazione di magenta.

Ho installato tool che usa magenta con il seguente comando:

```
pip install -qU pyfluidsynth pretty_midi
```

Dopo diverse prove con diversi compilatori python ho capito che magenta necessita di python 3.8 con tanto di librerie di sviluppo e per installarlo ho usato i seguenti comandi:

```
sudo add-apt-repository ppa:deadsnakes/ppa

sudo apt update

sudo apt install python3.8

sudo update-alternatives --install /usr/bin/python3 python3 /usr/bin/python3.8 1

sudo update-alternatives --config python3
```

```
sudo apt install python3.8-distutils
wget https://bootstrap.pypa.io/get-pip.py
python get-pip.py
sudo apt install python3.8-dev
pip install numpy==1.21.6
sudo apt install build-essential python3-dev libffi-dev libssl-dev
sudo apt-get install build-essential libasound2-dev libjack-dev portaudio19-dev
```

Ho installato strumenti di compilazione python con il seguente comando:

```
pip install --upgrade pip setuptools wheel
```

Ho installato le corrette versioni delle dipendenze di magenta:

```
pip install "setuptools<65"
pip install "numpy<1.23"
pip install "llvmlite<0.39" "numba<0.55"
```

E infine ho lanciato il seguente comando:

```
pip install magenta
```

Hello_world.py

Ho generato un file midi randomico e ho installato l'applicazione immidi per sentire il risultato. Magenta l'ho usato solo per esportare il file midi per testare che magenta funzionasse. Le note le ho generate randomicamente senza ausilio di intelligenza artificiale.

Modello.py

Ho continuato una melodia musicale con l'intelligenza artificiale a partire da un modello scaricato da internet a questo indirizzo:

http://download.magenta.tensorflow.org

Anche provando con diversi modelli il risultato cambia di poco.

Un progetto magenta chiamato **MusicVAE** è utile allo scopo in quanto genera intere composizioni musicali usando magenta.

MusicVAE è un modello di **variational autoencoder (VAE)** applicato alla musica. Un VAE è un tipo di rete neurale che apprende una rappresentazione compressa di un input complesso, come ad esempio una sequenza musicale, e può quindi generare nuovi esempi simili partendo da questa rappresentazione. Questo rende i VAEs particolarmente adatti alla generazione di dati creativi, come la musica.

Il link a vari modelli preaddestrati si trova a questo link:

https://github.com/magenta/magenta/tree/main/magenta/models/music_vae

Per usare questi modelli è necessario lanciare questo comando:

```
music_vae_generate \
--config=hierdec-trio_16bar \
--checkpoint_file=/home/pasquale/Scrivania/tesi/prove_magenta/magenta/models/music_vae/trio/checkpoints/hierdec-trio_16bar.tar \
--mode=sample \
--num_outputs=5 \
--output_dir=/home/pasquale/Scrivania/tesi/prove_magenta/magenta/models/music_vae/trio/checkpoints/generated
```

Questo comando genera 5 canzoni con 3 strumenti musicali: il basso, il piano e la batteria.

Approfondimento teorico

La musica può essere rappresentata in diversi formati per essere utilizzata in un contesto di reti neurali, ciascuna delle quali offre vantaggi specifici e presenta delle limitazioni, in funzione dell'obiettivo perseguito. Le tre principali rappresentazioni utilizzate per la musica in ambito computazionale sono: la rappresentazione simbolica tramite **MIDI**, la rappresentazione spettrale tramite **spettrogramma** e la rappresentazione dell'audio grezzo attraverso il **flusso di onde sonore**.

1. MIDI (Musical Instrument Digital Interface) - Rappresentazione Simbolica

Il formato **MIDI** rappresenta la musica in modo simbolico, conservando informazioni relative agli eventi musicali anziché al suono stesso. Nello specifico, il MIDI memorizza i seguenti parametri principali:

- **Nota On/Off**: il momento in cui una nota viene suonata o interrotta.
- **Altezza della nota**: la frequenza associata alla nota (ad esempio, C4, D#5, ecc.).
- **Velocità della nota**: l'intensità con cui una nota viene suonata.
- **Durata delle note**: il tempo in cui una nota è mantenuta attiva.

- **Informazioni sul canale e sullo strumento**: dettagli sull'instrumentazione utilizzata per ciascun suono.

Questa rappresentazione è particolarmente vantaggiosa per attività legate alla **composizione musicale**, in quanto consente una facile manipolazione delle note, della loro durata e intensità. Tuttavia, il formato **MIDI** non è in grado di catturare in modo completo il timbro e la qualità del suono, limitandosi alla rappresentazione delle note come eventi discreti. Inoltre, non riproduce fedelmente l'intensità e la variabilità dinamica dei suoni come sarebbe possibile in un contesto di registrazione audio.

2. Spettrogramma - Rappresentazione Spettrale

Lo spettrogramma è una rappresentazione visiva che descrive l'intensità delle frequenze sonore nel tempo. Questa rappresentazione viene ottenuta applicando la **Trasformata di Fourier** all'onda sonora, permettendo di separare il segnale in diverse componenti frequenziali. I parametri principali di uno spettrogramma sono:

- **Asse delle ascisse**: rappresenta il tempo.
- **Asse delle ordinate**: rappresenta la frequenza.
- **Colorazione o intensità**: indica l'ampiezza di ciascuna frequenza per un dato intervallo di tempo.

Uno degli aspetti distintivi dello spettrogramma è che consente di visualizzare in modo dettagliato l'evoluzione temporale delle frequenze, rendendolo utile per attività di **classificazione musicale**, **riconoscimento del genere** e **separazione delle sorgenti sonore**. Inoltre, trattandosi di una rappresentazione bidimensionale (tempo vs frequenza), è possibile applicare reti neurali convoluzionali (CNN), tecniche solitamente impiegate nell'elaborazione delle immagini, per estrarre caratteristiche e riconoscere pattern.

Tuttavia, una delle limitazioni principali di questa rappresentazione è la sua complessità computazionale. La creazione e l'elaborazione di spettrogrammi richiede un considerevole impegno di risorse, e l'approccio non è sempre ideale per la generazione musicale, in quanto non cattura la musicalità in modo simbolico, ma si concentra esclusivamente sulle caratteristiche spettrali.

3. Flusso di Onde Sonore - Rappresentazione dell'Onda Grezza

La **rappresentazione del flusso di onde sonore** implica l'uso dei dati grezzi audio, ovvero la sequenza dei campioni che descrivono l'andamento dell'onda sonora nel tempo. Ogni campione rappresenta una misura della pressione sonora in un dato istante. In questa rappresentazione, il

suono è riprodotto nella sua forma naturale, senza l'astrazione delle frequenze o dei parametri simbolici, e viene trattato come una sequenza temporale continua.

Questa modalità di rappresentazione è la più diretta e completa, poiché cattura tutte le informazioni timbriche, ritmiche e dinamiche del suono. La principale applicazione di questa rappresentazione è in compiti come il **riconoscimento vocale**, la **sintesi audio** o la **generazione musicale**. Tuttavia, l'uso dei dati grezzi audio comporta una notevole intensità computazionale. I file audio sono generalmente molto più pesanti rispetto a rappresentazioni simboliche o spettrali, e ciò può ostacolare l'elaborazione su larga scala, richiedendo grandi risorse di memoria e di potenza di calcolo.

Inoltre, l'analisi del flusso di onde sonore grezze richiede modelli complessi che possano gestire la natura sequenziale e continua del segnale, il che comporta una maggiore complessità nell'addestramento della rete neurale.

La scelta della rappresentazione più adatta dipende strettamente dall’obiettivo del progetto. Se l'intento è quello di **comporre musica** o manipolare le note, il formato **MIDI** risulta essere il più conveniente. Nel caso di **analisi del suono**, come il riconoscimento di generi musicali o la classificazione di eventi acustici, la rappresentazione tramite **spettrogramma** è altamente efficace. Infine, per compiti di **generazione musicale** o **sintesi audio**, l'uso del **flusso di onde sonore** rappresenta la scelta migliore, pur comportando un maggiore carico computazionale.

Nel contesto della **generazione automatica di musica**, progetti come **Magenta** di Google esplorano l'impiego di diverse rappresentazioni musicali per ottenere risultati creativi. Due delle principali modalità di rappresentazione della musica utilizzate da Magenta sono il formato **MIDI** e lo **spettrogramma**, con differenze sostanziali nei metodi di generazione e negli obiettivi finali. Questi approcci sono strettamente legati alle specificità dei modelli utilizzati, tra cui **MusicVAE**, **PerformanceRNN** e **Nsynth**.

Uno degli approcci prevalenti in **Magenta** per la generazione musicale è l’utilizzo del formato **MIDI**, una rappresentazione simbolica che codifica eventi musicali come note, durate, velocità e timbri. L'adozione di questo formato è motivata da diverse ragioni legate alla sua efficienza computazionale e alla sua capacità di rappresentare la musica a un livello astratto, concentrandosi sulle strutture musicali piuttosto che sui dettagli acustici.

- **Vantaggi del formato MIDI**:
    - **Rappresentazione simbolica**: MIDI non rappresenta direttamente il suono, ma gli eventi musicali come la pressione di una tasto su un pianoforte o la variazione della velocità di una nota. Ciò permette di modellare la musica a un livello superiore di

astrazione, dove l'attenzione è rivolta alle **relazioni tra le note**, alla **progressione armonica**, alla **melodia** e al **ritmo**. Questo rende il formato particolarmente adatto a compiti di **composizione musicale** automatica.

- **Efficienza computazionale**: Rispetto ad altri formati come i dati audio grezzi, i file MIDI sono molto più leggeri in termini di spazio di archiviazione e di elaborazione. Questo consente di gestire e generare musica in modo più rapido ed efficiente, senza necessitare di risorse computazionali particolarmente elevate.
- **Facilità di manipolazione e modifica**: Poiché MIDI rappresenta la musica in termini di eventi discreti, è facile intervenire su parametri specifici come l'intensità, la velocità, la durata e l'orchestrazione, rendendolo ideale per modelli di **generazione musicale** come **MusicVAE** e **PerformanceRNN**. Questi modelli, infatti, utilizzano i dati MIDI per **completare brani musicali** o **trasformare stili musicali**, consentendo la generazione di nuove composizioni a partire da tracce preesistenti.

Nel caso di **Nsynth**, un altro modello sviluppato all'interno del progetto Magenta, l'approccio è decisamente diverso. **Nsynth** sfrutta una rappresentazione spettrale per la generazione musicale, in particolare attraverso l'uso dello **spettrogramma** e di tecniche avanzate come **Wavenet**, un modello di deep learning che consente di generare suoni audio grezzi a livello di onda sonora.

- **Vantaggi dell'approccio basato su spettrogramma**:
  - **Rappresentazione spettrale**: Lo spettrogramma fornisce una rappresentazione visiva dell'intensità delle frequenze nel tempo, mostrando come il contenuto spettrale di un suono cambia nel corso della sua durata. Questa rappresentazione è utile per catturare la **complessità timbrica** dei suoni, consentendo a modelli come Wavenet di generare suoni che siano più realistici e dettagliati rispetto alla semplice manipolazione di note e durate, come avviene nel formato MIDI.
  - **Generazione audio grezzo**: A differenza del MIDI, che rappresenta la musica in termini simbolici, l'approccio basato sullo spettrogramma consente di generare suoni **realistici**. Nsynth, utilizzando **Wavenet**, è in grado di apprendere le caratteristiche timbriche degli strumenti musicali e generare nuove sonorità, combinando in modo innovativo suoni di strumenti diversi. Questo approccio è utile non solo per la generazione musicale, ma anche per la **creazione di nuovi suoni** attraverso la mescolanza e l'alterazione delle frequenze.
  - **Creazione di timbri unici**: L'uso del **modello Wavenet** permette a Nsynth di produrre suoni che non esistono nel mondo reale, ma che sono convincenti dal punto di vista timbrico, aprendo nuove possibilità nella creazione sonora. Wavenet lavora

su una sequenza di valori audio grezzi, generando onde sonore che possono essere combinate in modo complesso per ottenere risultati mai sentiti prima.

Nel contesto della creazione di **brani musicali semplici** destinati a **ragazzi con disabilità**, è fondamentale scegliere un modello di generazione musicale che possa produrre composizioni facilmente comprensibili, accessibili e stimolanti. Due approcci principali che possono essere considerati per questo scopo sono l'uso del modello **Melody RNN** di **Magenta** e la creazione di un **modello personalizzato basato su canzoni ballabili**.

Utilizzo di Melody RNN per la Generazione di Melodie Semplici

Melody RNN, uno dei modelli di generazione musicale sviluppati nell'ambito del progetto **Magenta** di Google, è stato progettato per generare **melodie semplici** e **orecchiabili**. Questo modello è basato su **reti neurali ricorrenti (RNN)** ed è in grado di generare sequenze musicali che proseguono a partire da una melodia di partenza, producendo brani musicali brevi e lineari, caratterizzati da **una struttura chiara e facilmente fruibile**.

1. **Semplicità delle Melodie**: Melody RNN si distingue per la capacità di generare melodie **semplici e ripetitive**, che sono particolarmente adatte per un pubblico giovane o con **difficoltà cognitive**. Le melodie create tramite questo modello tendono ad essere lineari e facilmente comprensibili, evitando complessità armoniche o ritmiche che potrebbero risultare difficili da seguire.
2. **Controllo della Complessità**: Un aspetto fondamentale di Melody RNN è la possibilità di **regolare la complessità** delle melodie generate. È possibile modulare il livello di **ripetizione** e di **variabilità ritmica**, adattando i brani alle specifiche esigenze del pubblico di riferimento. Questo permette di ottenere **composizioni musicali facili da seguire** e particolarmente **adatte per attività educative o terapeutiche**.
3. **Adattabilità a Diversi Stili Musicali**: Sebbene il modello Melody RNN sia versatile, esso può essere facilmente adattato per produrre **brani di generi musicali semplici**, come canzoni pop, melodie allegre o ritmi ballabili, che risultano particolarmente coinvolgenti per i ragazzi. Questi tipi di composizioni sono utili per stimolare attività ludiche o motorie, come il **ballo** o la **movimentazione**, che possono essere terapeutiche per i bambini con disabilità.

Creazione di un Modello Personalizzato Basato su Canzoni Ballabili

In alternativa all'uso di Melody RNN, è possibile **creare un modello personalizzato** basato su **canzoni ballabili** o generi musicali specifici, in modo da ottimizzare la musica per un pubblico con

particolari esigenze motorie o cognitive. Questo approccio prevede l'addestramento di una rete neurale su un **dataset personalizzato** che include esclusivamente canzoni caratterizzate da **ritmi regolari**, **melodie semplici** e **strutture facilmente riconoscibili**.

1. **Personalizzazione del Dataset**: Creare un modello personalizzato permette di addestrare la rete neurale utilizzando esclusivamente **brani ballabili** o di genere musicale con caratteristiche ritmiche regolari. Ciò consente di ottenere **musica più mirata** e **adatta a stimolare attività fisiche**, come il ballo, che è particolarmente utile per ragazzi con **disabilità motorie** o **difficoltà di coordinazione**.
2. **Adattamento alle Esigenze Terapeutiche e Educative**: La musica generata da un modello personalizzato può essere specificamente progettata per **stimolare la coordinazione motoria**, l'attenzione e altre competenze cognitive. L'uso di **ritmi regolari** e melodie **ripetitive** è particolarmente indicato per **facilitare la comprensione e la partecipazione attiva** in attività educative o terapeutiche. Inoltre, questa musica può essere adattata per rispondere alle **preferenze individuali** degli utenti, ad esempio, scegliendo canzoni con tempi lenti o allegri in base alle necessità.
3. **Generazione di Timbriche Coinvolgenti**: Un altro vantaggio di un modello personalizzato è la possibilità di creare **musica che incoraggi l'interazione fisica** e sensoriale, utilizzando suoni che stimolino il movimento e l'espressione corporea. Per esempio, brani con **ritmi allegri e tempi veloci** possono favorire attività motorie come il ballo, mentre melodie più tranquille potrebbero essere utili per momenti di rilassamento o stimolazione sensoriale.

Nel contesto della creazione di **musica semplice e coinvolgente** per **ragazzi con disabilità**, **Melody RNN** rappresenta una scelta valida per generare melodie accessibili e facilmente fruibili. La sua capacità di produrre **brani musicali ripetitivi** e semplici lo rende particolarmente adatto a compiti educativi o terapeutici, dove la chiarezza e la regolarità della musica sono essenziali. Tuttavia, la creazione di un **modello personalizzato basato su canzoni ballabili** potrebbe essere un'opzione ancora più mirata, permettendo di ottimizzare la musica in base alle **esigenze specifiche** del pubblico, come la stimolazione motoria, la coordinazione e l'interazione fisica. La scelta tra i due approcci dipende dalla necessità di personalizzazione, dall'obiettivo educativo e terapeutico e dalla disponibilità di risorse computazionali e di tempo per l'addestramento del modello.

Musegan

Musegan e Magenta sono entrambi progetti avanzati nel campo della generazione musicale automatica, ma presentano differenze chiave nel loro approccio e nelle loro capacità. Il sito di musegan è https://hermandong.com/musegan/.

Mentre **Magenta** è più orientato alla ricerca e offre maggiore flessibilità nella manipolazione delle melodie, **Musegan** si distingue per la sua capacità di generare musica multitraccia in modo autonomo, creando composizioni musicali complete con diversi strumenti. Se l'obiettivo è generare accompagnamenti musicali completi o brani con più strumenti in un singolo passo, Musegan potrebbe essere la scelta migliore. Se invece si cerca una piattaforma più versatile e focalizzata sulla composizione di melodie e arrangiamenti in vari generi, **Magenta** potrebbe essere più adatto. Supponendo di creare prima la musica Musegan potrebbe essere l'approccio più semplice, ma per lasciarsi aperte le porte alle sperimentazioni e ad eventuali approcci alternativi userò magenta.

Voce cantata

Prospettive per una voce cantata professionale si potrebbe avere con tacotron https://github.com/NVIDIA/tacotron2 (gpu) e wavenet , ma ho trovato difficoltà nello scaricare il modello di tacotron in quanto richiede abilitazione e quindi ho rinunciato allo spike.

Composizione della musica e come usare AI per comporla

Musica prima, testo poi (Approccio tradizionale e più semplice da automatizzare)

L'approccio in cui la **musica precede il testo** è il più tradizionale e diffuso, particolarmente nei generi musicali come **pop**, **rock**, **jazz** e in parte nella **musica classica**. In questo caso, il compositore si concentra inizialmente sulla creazione della parte musicale, che include la melodia, l'armonia e la ritmica, e successivamente scrive il testo per adattarsi alla musica stessa. Questo approccio è particolarmente vantaggioso quando la **musica è il focus centrale** del brano, come nel caso di molte canzoni pop, dove la melodia è spesso l'elemento principale. Il testo, in questo caso, viene progettato per rispecchiare il tono emotivo e ritmico della musica.

Aspetti creativi

Dal punto di vista creativo, la **musica prima** offre una maggiore libertà nella composizione melodica e armonica. Una volta che la musica è stata scritta, il compositore può **adattare** il testo in base alla melodia e al ritmo, cercando di enfatizzare determinati temi o emozioni che la musica stessa evoca. La metrica della melodia (il numero di battute per frase musicale) guida la struttura del testo, che viene "modellato" per adattarsi alla musica già esistente. Questo processo può essere

particolarmente utile quando il brano ha una **melodia forte e memorabile**, e il testo deve adattarsi a una struttura musicale già definita.

Aspetti informatici

Dal punto di vista informatico, l'approccio di "musica prima, testo poi" si presta particolarmente bene all'automazione, poiché le problematiche principali si concentrano sulla generazione musicale. La generazione automatica di musica, tramite modelli di machine learning come **MusicVAE** (Magenta), è relativamente semplice, poiché il modello può concentrarsi esclusivamente sulla **creazione di sequenze melodiche, armoniche e ritmiche**, senza dover considerare variabili complesse legate all'adattamento di un testo.

Un modello di **intelligenza artificiale** in questo caso deve:

- **Definire una tonalità** per la musica, che diventa il punto di partenza per la creazione delle sequenze melodiche.
- **Generare una melodia** che segua una struttura armonica predefinita o che si adatti a uno stile musicale specifico.
- **Adattare la ritmica** della musica in modo che il testo successivamente scritto possa essere facilmente adattato alla struttura ritmica e melodica della musica generata.

Un esempio di applicazione di questo approccio potrebbe essere l'uso di **vocoder** per la parte vocale, che può essere facilmente adattato alla musica generata in quanto non richiede una sincronizzazione complessa tra **voce e struttura musicale**. In altre parole, il vocoder può essere usato per modificare la voce affinché si adatti alla musica generata, ma non c'è necessità di adattare la musica stessa alla voce.

Testo prima, musica poi (Maggiore complessità nell'automazione)

In questo approccio, il **testo viene scritto prima della musica**, e la musica viene successivamente creata per accompagnare le parole. Questo approccio è più comune in **musicali**, **opera lirica** e in alcune canzoni pop, dove il messaggio o la narrazione del testo è considerato fondamentale. Il compositore, dopo aver scritto le parole, si dedica alla composizione musicale che meglio si adatta al testo, al suo significato e alla sua metrica.

Aspetti creativi

Dal punto di vista creativo, il testo svolge un ruolo centrale, poiché il compositore si preoccupa prima di trasmettere un messaggio o una narrazione attraverso le parole. Una volta completato il

testo, il passo successivo è creare una musica che rispecchi il **tono emotivo**, **il ritmo** e la **struttura metrica** del testo. Questo processo implica che la musica debba essere adattata alle specifiche esigenze del testo, come ad esempio:

- La scelta del **genere musicale** che meglio si adatti al contenuto emotivo e tematico del testo (ad esempio, una canzone triste può richiedere una tonalità minore e un ritmo lento).
- La definizione della **ritmica** per il testo, che può essere variabile a seconda della velocità e della cadenza delle parole.
- L'adattamento della **tonalità** della musica per garantire che la voce cantata possa essere eseguita comodamente.

Aspetti informatici

Dal punto di vista informatico, generare **musica che si adatti a un testo scritto** presenta una serie di sfide significative. In particolare, il processo di adattamento musicale al testo richiede l'analisi e l'integrazione di vari fattori che non sono immediatamente risolvibili tramite modelli automatici:

- **Scelta del genere musicale**: Il modello dovrebbe essere in grado di **comprendere** il contenuto del testo (ad esempio, temi come la tristezza, la gioia, la speranza) e generare un accompagnamento musicale che rispecchi l'emozione evocata dalle parole. Ciò richiede una certa comprensione semantica del testo da parte del modello.
- **Adattamento ritmico**: Il modello deve essere in grado di **sincronizzare** la musica con la metrica del testo, tenendo conto del numero di sillabe, degli accenti e della struttura prosodica del linguaggio. La musica deve essere in grado di seguire o enfatizzare il ritmo delle parole, un compito che diventa complesso quando il testo presenta variazioni ritmiche.
- **Tonalità e intonazione**: La tonalità della musica deve essere scelta in modo da non solo rispecchiare l'emozione del testo, ma anche consentire che la **voce cantata** si esprima senza difficoltà. Inoltre, il modello dovrebbe garantire che la **tonalità della musica** sia compatibile con la vocalità umana, considerando intervalli melodici che siano adeguati a una performance vocale naturale.

Un altro problema significativo è che, nel caso in cui il testo contenga una **storia complessa**, la musica deve essere in grado di **evolversi** nel corso del brano per riflettere i cambiamenti emotivi o narrativi del testo. Ciò implica la creazione di una **musica dinamica**, che possa rispondere ai cambiamenti tematici nel testo.

Scrittura parallela (Approccio complesso)

La **scrittura parallela** è un approccio in cui **musica e testo vengono scritti simultaneamente**, influenzandosi a vicenda durante il processo creativo. Questo approccio è il più complesso dal punto di vista informatico, poiché richiede un'interazione continua tra la generazione della musica e quella del testo. In pratica, il compositore scrive la musica e il testo insieme, cercando di ottenere una sinergia perfetta tra i due elementi.

Aspetti creativi

Nel processo di scrittura parallela, **musica e testo si influenzano reciprocamente**. La struttura musicale può suggerire determinate scelte stilistiche e tematiche per il testo, mentre il testo può determinare variazioni nella musica. Questo approccio è tipico in **composizioni complesse** come **musicali** o **opera lirica**, dove la narrazione musicale e testuale è fondamentale. Durante la scrittura parallela, il compositore può decidere di **modificare il testo** in base alle evoluzioni della musica, o viceversa, facendo in modo che il risultato finale sia una composizione ben bilanciata.

Aspetti informatici

Dal punto di vista informatico, **automazionare** la scrittura parallela è particolarmente difficile, poiché impone la necessità di **iterazioni continue** tra la generazione del testo e quella della musica. Il modello dovrebbe essere in grado di **adattare sia la musica che il testo** in tempo reale, mentre ciascuno dei due elementi influenza l'altro. Ciò implica che un sistema informatico dovrebbe essere in grado di:

- Analizzare la **struttura narrativa** e le **emozioni** del testo, mentre genera una musica che rispecchi questi aspetti.
- Adattare il **ritmo della musica** in base al testo e viceversa, per garantire una coerenza metrica ed emotiva.
- **Modificare il testo** in base agli sviluppi musicali, o modificare la musica in base al testo, creando una composizione coerente.

Questo tipo di approccio potrebbe essere intrapreso da un compositore magari con strumenti come https://acestudio.ai.

Priorità della Musica nella Composizione Automatizzata

Nel contesto della mia applicazione, anche se il **testo** è il principale elemento espressivo della composizione, risulta più vantaggioso adottare l'approccio in cui la **musica viene generata prima del testo**. Sebbene in alcuni generi musicali il testo costituisca il nucleo centrale dell'opera, la

generazione automatica della musica si presta a essere affrontata con maggiore efficacia e meno complessità computazionale se precede la scrittura del testo.

Dal punto di vista informatico, la **generazione automatica della musica** è più facilmente realizzabile in prima istanza, poiché si concentra su aspetti come la **melodia**, l'**armonia** e il **ritmo** senza dover tenere conto di variabili complesse legate all'adattamento del testo. Questo approccio consente di creare una base musicale che funzioni come struttura portante del brano, sulla quale il testo potrà successivamente essere inserito, adattato e modellato.

Inoltre, la musica generata può essere facilmente armonizzata con il testo attraverso tecniche come l'uso di un **vocoder** o altre metodologie di **sincronizzazione vocale**, che permettono di adattare la voce al ritmo e alla tonalità musicali. Pertanto, la scelta di generare prima la musica consente di semplificare il processo creativo e computazionale, facilitando l'integrazione del testo senza dover risolvere la complessità dell'adattamento reciproco tra musica e parole in un unico passo.

Requisiti

Requisiti Funzionali

1. **Interfaccia Utente Accessibile**
    - Design semplice e intuitivo.
    - Opzioni di personalizzazione dei colori e dei caratteri.
2. **Registrazione Utente**
    - Form di registrazione per raccogliere informazioni su età, interessi e problematiche.
3. **Generazione di Brani Musicali**
    - Brani generati in modo predefinito tramite un backoffice.
    - Possibilità di scaricare brani già creati dagli utenti.
4. **Riconoscimento Facciale**
    - Monitoraggio delle reazioni emotive dell'utente tramite la telecamera.
    - Interventi in tempo reale basati su segnali di frustrazione.
5. **Fasi di Apprendimento**
    - Prima fase: presentazione di parole/frasi con supporti visivi e sonori.
    - Seconda fase: integrazione delle parole in canzoni scaricate dal backoffice.
6. **Feedback e Correzione**
    - Feedback immediato sulla pronuncia durante la fase di apprendimento.

- Suggerimenti per migliorare e motivare durante il canto.

7. **Messaggi Motivazionali**
    - Integrazione di messaggi incoraggianti in caso di difficoltà.
8. **Modalità di Gioco**
    - Attività ludica per rinforzare l’apprendimento attraverso il canto.

Requisiti Non Funzionali

1. **Accessibilità**
    - Compatibilità con strumenti assistivi (screen reader, ecc.).
    - Supporto per diverse lingue e dialetti.
2. **Prestazioni**
    - Risposta rapida dell'interfaccia e dei sistemi di riconoscimento emotivo.
    - Minimizzazione dei tempi di attesa per il download delle canzoni.
3. **Sicurezza**
    - Protezione dei dati personali raccolti durante la registrazione.
    - Opzioni di privacy per l'uso della telecamera.
4. **Scalabilità**
    - Capacità di gestire un numero crescente di utenti e contenuti musicali.
5. **Usabilità**
    - Test di usabilità con utenti finali per garantire che l'app sia facilmente navigabile.
    - Documentazione chiara per utenti e genitori.

Requisiti Tecnici

1. **Piattaforma**
    - Compatibilità con dispositivi Android per la parte di front-end.
2. **Tecnologie di Intelligenza Artificiale**
    - Algoritmi per generazione musicale da utilizzare nel backoffice.
3. **Riconoscimento Facciale**
    - Integrazione di librerie di riconoscimento facciale per l'analisi delle emozioni.
4. **Database**
    - Archiviazione sicura dei dati utente e dei brani musicali generati.

5. **Sistema di Backoffice**
    - Interfaccia per la creazione e gestione delle canzoni.
    - Funzionalità per caricare brani generati nel sistema e renderli disponibili per il download.
6. **Docker**
    - La soluzione deve essere incapsulata in un container Docker per garantire la portabilità, la scalabilità e un ambiente di esecuzione consistente su diverse piattaforme e ambienti (sviluppo, test, produzione). L'applicazione e tutte le sue dipendenze devono essere raccolte in un'immagine Docker tramite un Dockerfile, che definisce in modo esplicito l'ambiente di esecuzione, le librerie necessarie e le configurazioni.

Analisi

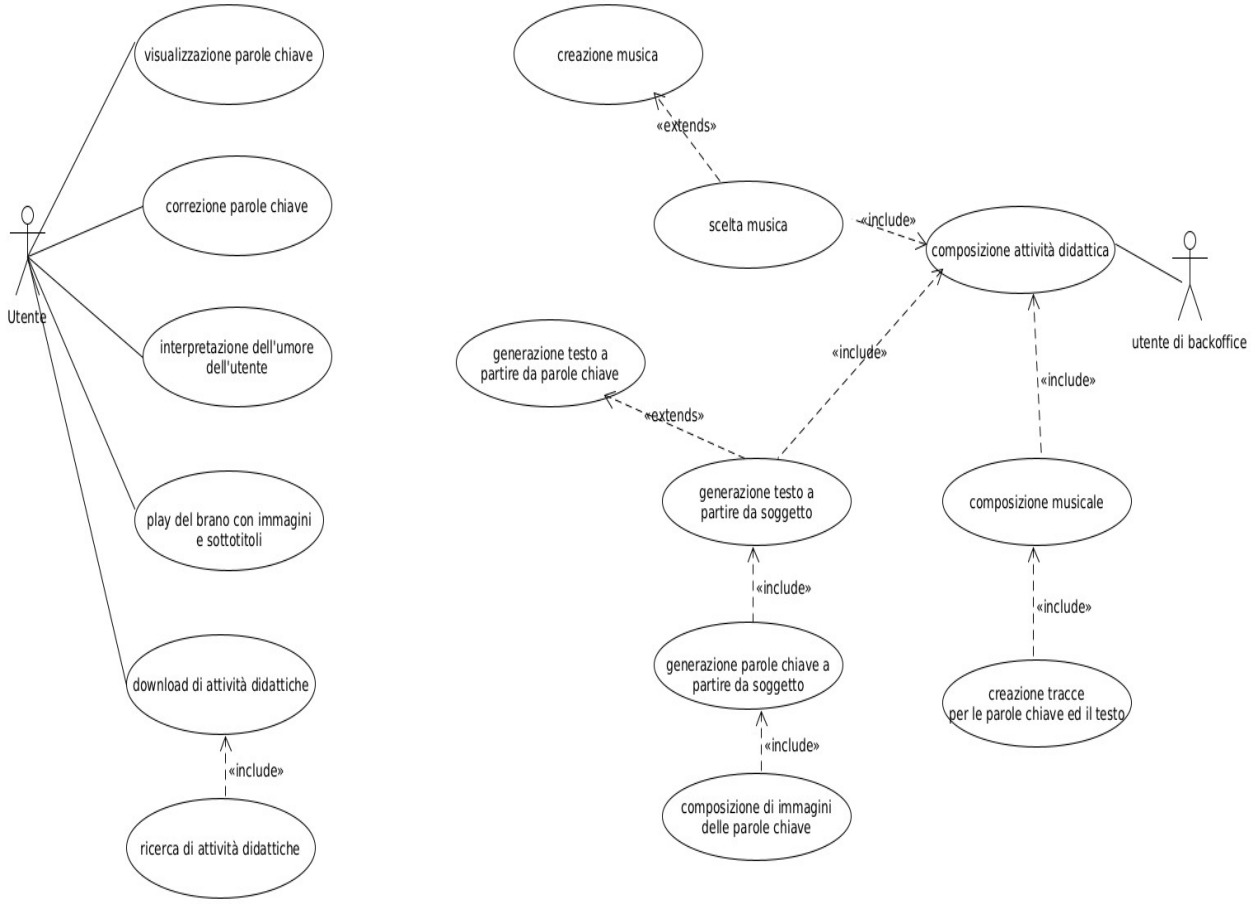


Figura 3: Analisi del progetto completo

Il presente diagramma illustra i vari casi d'uso associati al progetto finale. Considero particolarmente significativi i primi quattro casi d'uso dell'utente, in quanto rappresentano le funzionalità minime necessarie per l'utilizzo del progetto. È fondamentale selezionare tecnologie adeguate fin dall'inizio, al fine di garantire la possibilità di estendere il progetto in futuro senza la necessità di ricominciare da capo.

Il download delle attività didattiche potrebbe essere visto anche solo come un abilitazione all’uso delle attività didattiche per l’utente loggato.

Un attività didattica comprende:

- testo del brano musicale
- parole chiave del testo
- streaming cantato del brano
- streaming non cantato del brano

- timeline delle parole chiave
- associazione tra parole chiave e immagini del brano musicale
- timeline del testo

Un possibile caso d’uso non espresso può essere un sistema di pagamento al download dell’attività didattica. Per questo caso d’uso utilizzerei paypal in quanto di facile integrazione e riconosciuto da qualunque utente. Inoltre se questa idea di progetto dovesse mai portare ad aprire un’associazione no profit ci sarebbero ulteriori sconti sulle tariffe applicate ai pagamenti.

Durante la riunione con la cooperativa sociale "Il Margine" di Settimo Torinese, nell'ambito del progetto "Il Ponte", sono emerse alcune obiezioni e proposte migliorative riguardo l'idea iniziale del progetto.

In primo luogo, è stata sollevata l'obiezione sull'uso di canzoni esistenti all'interno del progetto, con l'intento di facilitare la memorizzazione da parte degli utenti. Tuttavia, questa proposta è stata respinta per due motivi principali: da un lato, l'uso di brani già protetti da diritti d'autore avrebbe comportato il pagamento delle relative royalty alla SIAE, aumentando significativamente il costo finale del prodotto; dall'altro, le canzoni esistenti potrebbero non rispondere agli obiettivi educativi prefissati, poiché il testo potrebbe non essere congruente con gli scopi del progetto. Come possibile soluzione a questa problematica, è stata suggerita l'idea di far ascoltare la canzone agli utenti prima dell'esercizio pratico, in modo da permettere loro di familiarizzare con il brano in anticipo. Inoltre, si è proposto di offrire la possibilità di scaricare la canzone, consentendo così di ascoltarla anche al di fuori del contesto dell'applicativo.

In secondo luogo, è stato avanzato un suggerimento riguardo alla creazione delle attività didattiche, sottolineando l'importanza di prestare particolare attenzione alla scelta dei contenuti. Se le attività si basano su frasi di vita quotidiana, come quelle relative all'amore, alla cucina o ad altri temi di interesse comune, è possibile attrarre maggiormente l'utente finale, stimolando un maggiore coinvolgimento.

Infine, è stato proposto di includere attività didattiche che abbiano anche una componente ludica e fisica, come ad esempio giochi di movimento (tipo "gioca juer" di Claudio Cecchetto) o l'utilizzo di canzoni facilmente ballabili, come la bachata. L'inclusione di attività che prevedano il ballo potrebbe favorire un'esperienza più dinamica e coinvolgente per gli utenti, aumentando l'efficacia del progetto stesso.

In sintesi, le obiezioni e i suggerimenti emersi dalla cooperativa "Il Margine" sono stati fondamentali per orientare il progetto verso una direzione più mirata e adatta alle necessità degli utenti finali, ottimizzando al contempo gli aspetti educativi e ludici.

Durante la riunione con la cooperativa sociale "Il Margine" di Settimo Torinese, è stato valutato positivamente il prodotto proposto nell'ambito del progetto "Il Ponte". In particolare, è emersa l'intenzione di condurre dei test con utenti finali per verificare l'efficacia del prodotto in contesti reali. A tal proposito, è stato suggerito di somministrare un questionario a seguito dei test, al fine di raccogliere feedback sul loro esito e comprendere meglio le impressioni degli utenti riguardo all'esperienza offerta dal prodotto. Questo approccio consentirebbe di ottenere dati concreti e utili per eventuali miglioramenti, rendendo il progetto ancora più aderente alle esigenze degli utenti finali. Tuttavia, i test saranno realizzati solo se i responsabili della cooperativa decideranno di aderire a questa proposta.

Design

Per quanto concerne il design, ho condotto una ricerca online al fine di individuare uno strumento adeguato ai miei obiettivi. Dopo un'attenta valutazione, ritengo di aver trovato la soluzione idonea in Webflow. Intendo approfondire la mia conoscenza di questa piattaforma attraverso la sua accademy, per verificare se essa disponga delle funzionalità necessarie a rappresentare visivamente le mie idee.

Webflow si presenta come uno strumento relativamente sofisticato per le mie esigenze, ma la sua limitazione principale risiede nelle modalità di esportazione dei contenuti: esse sono infatti ridotte a screenshot dello schermo o all'acquisto di un abbonamento che consenta l'esportazione del codice. Pertanto, intendo esplorare un'alternativa considerata più user-friendly, denominata Canva.

Il font scelto è **Arial** in quanto il font sans-serif è semplice e chiaro e quindi molto leggibile. Le dimensioni dei titoli sono di 26pt e del testo normale 18 pt.

Schermate

1. Home Page

- **Sfondo**: Colore chiaro (azzurro o verde pastello).
- **Logo**: Al centro in alto, grande e colorato.
- **Titolo**: “Karaoke Inclusivo” sotto il logo, font grande e leggibile.
- **Menu di Navigazione**: Pulsanti per:

- Karaoke/Apprendimento
- Scarica Unità Didattiche
- Supporto
- Login/registrazione/profilo utente

Figura 4: home page ipotizzata

2. Schermata di Karaoke e Esecuzione Canzoni/Apprendimento

- **Sfondo**: Colore vivace (giallo o arancione).
- **Titolo**: “Inizia a Cantare” in alto.
- **Visualizzazione del Testo**: Grande, con colori alternativi per le parole.
- **Controlli Musicali**:
    - **Pulsanti**: Play, Pausa, Rallenta, Messaggi Motivazionali.

- **Selezione Canzone**: Un menu a discesa per scegliere i brani disponibili.

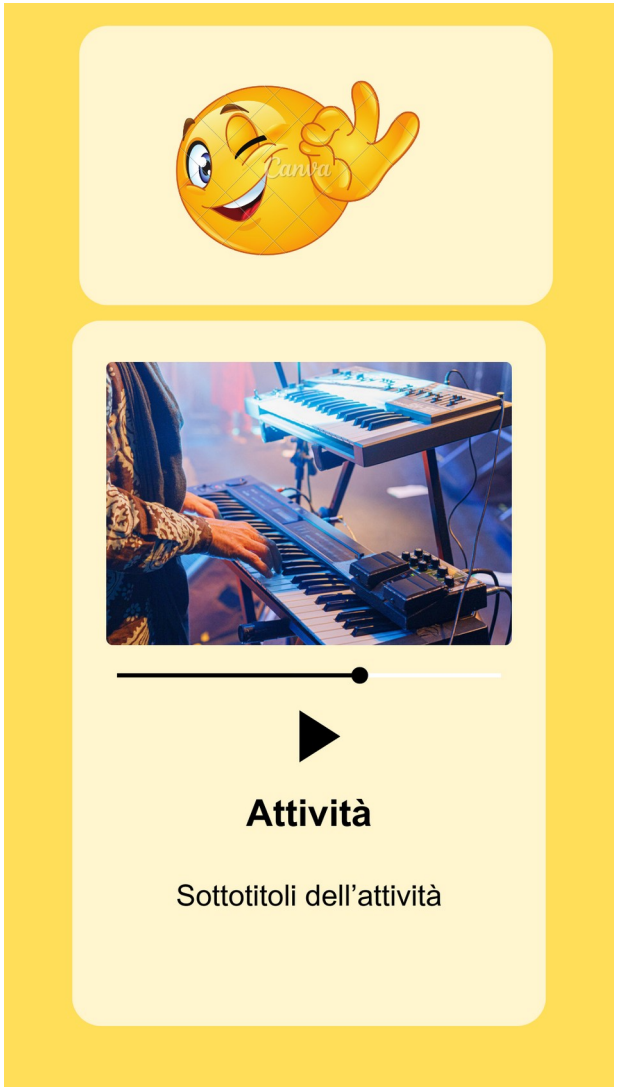


Figura 5: schemata ipotizzata del karaoke in esecuzione

3. Schermata di Scaricamento Unità Didattiche

Ricerca e download delle unità didattiche in base anche alle preferenze dell'utente

- **Sfondo**: Colore chiaro (azzurro o verde pastello).
- **Titolo**: "Scarica Unità Didattiche" in alto.
- **Barra di Ricerca**: Area per cercare unità didattiche specifiche.
- **Elenco di Unità Didattiche**:
    - Visualizzazione in forma di schede, con titoli e descrizioni brevi.
    - Pulsanti di "Scarica" accanto a ciascuna unità.

4. Profilo Utente

- **Sfondo**: Colore chiaro (azzurro o verde pastello).
- **Titolo**: “Il Tuo Profilo” in alto.
- **Informazioni Utente**: Visualizza età, interessi e preferenze.
- **Pulsante di Modifica**: Grande e accessibile.
- **Salvataggio**: Pulsante “Salva Modifiche” in fondo.

5. Sezione Supporto

- **Sfondo**: Colore chiaro (azzurro o verde pastello).
- **Titolo**: “Hai Bisogno di Aiuto?” in alto.
- **FAQ e Contatti**:
    - Domande frequenti elencate con caselle cliccabili.
    - Informazioni di contatto.
- **Pulsante di Contatto**: “Contattaci” ben visibile.

6. Sezione di Login/registrazione standard

Una volta loggato l’utente avrà a disposizione la possibilità di scaricare nuove unità didattiche.

Layout Generale
Sfondo: Colore chiaro (azzurro o verde pastello) per un’atmosfera accogliente.

Architettura e attività proposte

Il progetto prevede l'utilizzo di Django con due applicazioni: una dedicata agli utenti di backoffice e l'altra per gestire le chiamate del front-end. Inizialmente, sarà sviluppata esclusivamente l'applicazione per il front-end. L'autenticazione degli utenti avverrà tramite token di Django Rest Framework, implementata però in una fase successiva.

Front-end sviluppato con React, libreria di componenti chakra-ui e **react-a11y** per rendere l’html accessibile.

Studio del framework magenta, tensor-flow e altri strumenti utili per la generazione della componente musicale.

Per una POC mi concentrerei principalmente sulle prime 2 schermate e continuerei a fare ricerca sulla creazione di canzoni. Questo è il link del trello che mi sono immaginato al momento:

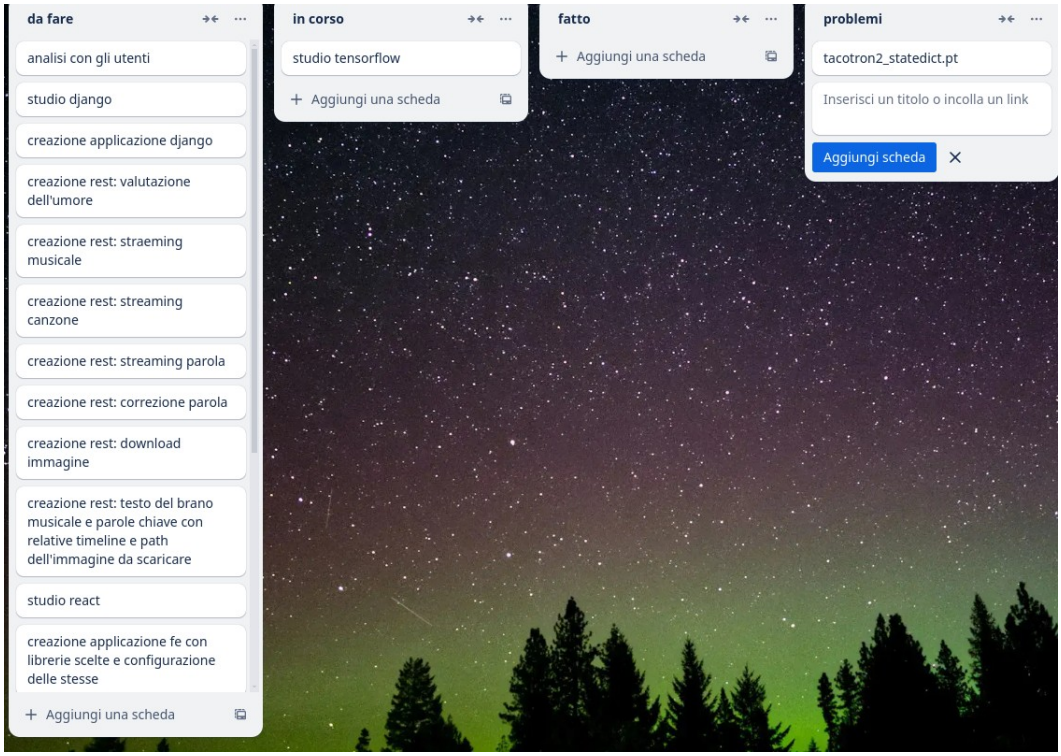


Figura 6: trello, organizzazione del lavoro

Il link al trello è questo: https://trello.com/b/OJ2TIOTh/tesi

Implementazione

Studio di Django

Trovandomi a dover gestire diversi ambienti di lavoro, specialmente in contesti complessi come l'intelligenza artificiale, `venv` non mi è sufficiente, soprattutto quando si ha bisogno di gestire versioni diverse delle librerie, ognuna delle quali potrebbe richiedere configurazioni specifiche per ogni progetto.

In scenari come quelli dell'**intelligenza artificiale** (AI) e del **machine learning** (ML), dove le librerie e i framework sono molto specifici e spesso dipendono da versioni precise di Python e di pacchetti esterni (come `TensorFlow`, `PyTorch`, `NumPy`, `scikit-learn`, e così via), strumenti come `conda` sono necessari.

Dopo aver scaricato miniconda con il seguente comando:

```
wget https://repo.anaconda.com/miniconda/Miniconda3-latest-Linux-x86_64.sh
```

Il seguente comando crea un environment dove è installato python3.8:

```
conda create --name ai_project python=3.8
```

Per attivare l’environment invece si usa:

```
conda activate ai_project
```

dove ai_project è il nome dell’environment che si vuole attivare.

Volendo installare magenta senza pip si potrebbe usare conda-forge ma non riesco a capire quale versione di magenta è pacchettizzata. Anche se cercando magenta con il seguente comando

```
conda search -c conda-forge magenta
```

non ho trovato risultati quindi credo che questa strada non sia la strada giusta.

Con questo comando ho creato il progetto:

```
django-admin startproject karaoke_be .
```

Dopodichè ho creato le api con questo comando:

```
python3 manage.py startapp karaokeapi
```

Per lanciare l’applicazione linkata correttamente in settings.py basta lanciare il seguente comando

```
python3 manage.py runserver 0.0.0.0:8000
```

e django lancerà un http server sulla porta 8000.

Per interagire con le tabelle di django è necessario creare un modello e un Manager del modello. Dopodichè si lancerà il comando:

Per lanciare l’applicazione linkata correttamente in settings.py basta lanciare il seguente comando

```
python3 manage.py makemigrations karaokeapi
```

dove karaokeapi è l’applicazione da migrare per creare uno script che modificherà il db di django.

Per eseguire lo script creato si lancerà il comando:

```
python3 manage.py migrate
```

Per creare un superuser sul db di django è necessario eseguire il seguente comando:

```
python3 manage.py createsuperuser
```

A questo punto dopo aver registrato il modello in admin.py sotto l’url 127.0.0.1:8000/admin si vedrà il nuovo modello UserProfile.

Per capire come funziona Django rest framework ho creato un api get e post. Per fare ciò ho usato API Views e ho creato un serializer che mi valida l’input della post. Ho anche mappato una url alla view appena creata.

Ho definito un file serializer.py dove metterò tutti i serializer necessari per le view dell’applicativo.

Ho sperimentato anche delle api che mettono a disposizione un CRUD di un entità. Esse hanno il nome di Viewset Api.

Ho creato un’api di login usando Api view estendendo una classe di default di Django chiamata ObtainAuthToken e settando un default renderer preso dai settings dell’applicazione.
l'autenticazione avviene attraverso l'invio di un nome utente e una password, con la restituzione di un token di autenticazione. Questo token, una volta ottenuto, può essere utilizzato per accedere alle risorse protette nell'applicazione. Nonostante questa soluzione offra un livello di sicurezza, essa presenta alcune vulnerabilità intrinseche, in particolare in relazione alla protezione del token durante la trasmissione attraverso la rete.

Il principale rischio associato all'utilizzo di token in un contesto non sicuro (HTTP anziché HTTPS) è la possibilità di intercettazione del token stesso da parte di un attaccante che si trova sulla stessa rete dell'utente (come nel caso di una rete Wi-Fi non protetta). In uno scenario di attacco MitM, l'attaccante può intercettare il traffico HTTP non criptato e, conseguentemente, rubare il token di autenticazione. Una volta in possesso del token, l'attaccante può impersonare l'utente e accedere alle risorse protette dell'applicazione, minando la sicurezza del sistema.

Per rafforzare la sicurezza del sistema di autenticazione e ridurre il rischio di attacchi MitM, è fondamentale adottare alcune best practices in materia di sicurezza delle applicazioni web. Di seguito vengono descritti alcuni approcci che possono migliorare significativamente la protezione dell'autenticazione basata su token.

Il primo passo fondamentale per proteggere i dati sensibili durante la trasmissione è l'adozione del protocollo HTTPS (HyperText Transfer Protocol Secure). HTTPS garantisce che le comunicazioni tra il client e il server siano criptate tramite SSL/TLS, impedendo che i dati, inclusi i token di autenticazione, possano essere intercettati o modificati da attaccanti durante il loro transito attraverso la rete. In un sistema di autenticazione, HTTPS è essenziale per evitare che un token venga rubato in un attacco MitM, dato che il traffico criptato non può essere facilmente decrittato senza il giusto certificato.

Per migliorare ulteriormente la sicurezza del sistema di autenticazione, è possibile passare a un'implementazione basata su JSON Web Token (JWT). I JWT sono token criptati che contengono informazioni aggiuntive, come il payload e i claim, che possono includere l'identità dell'utente, i

permessi e altre informazioni contestuali. Uno degli aspetti principali che rende i JWT vantaggiosi è la possibilità di definire una scadenza per il token (tramite il campo "exp"), limitando così il tempo durante il quale il token rimane valido. In caso di furto di un token, l'attaccante avrà accesso solo per un periodo limitato.

Inoltre, i JWT offrono vantaggi in termini di scalabilità e gestione dei permessi, in quanto è possibile includere all'interno del token informazioni critiche per l'accesso, riducendo la necessità di interrogare un database ogni volta che un utente fa una richiesta.

Un altro aspetto cruciale dell'adozione di JWT è la validazione dei claim. I claim sono dichiarazioni che il server può inserire nel token, come il tempo di scadenza (`exp`), l'emittente (`iss`), il destinatario (`aud`), ecc. La verifica di questi claim da parte del server durante ogni richiesta consente di accertare l'autenticità e la validità del token prima che l'utente possa accedere alle risorse protette. Inoltre, l'uso di una firma digitale (HMAC o RSA) sul token garantisce che il token non sia stato manomesso durante il suo transito.

L'uso di una semplice API di login con token, come quella realizzata estendendo la classe `ObtainAuthToken` di Django REST Framework, sebbene utile, presenta dei rischi di sicurezza accettati in quanto poc del prodotto finale, quest'ultimo potrà usare djangorestframework-simplejwt per l'implementazione della stessa.

Per quanto riguarda il db sto usando SqlLite di default in Django e lo schema ER è il seguente:

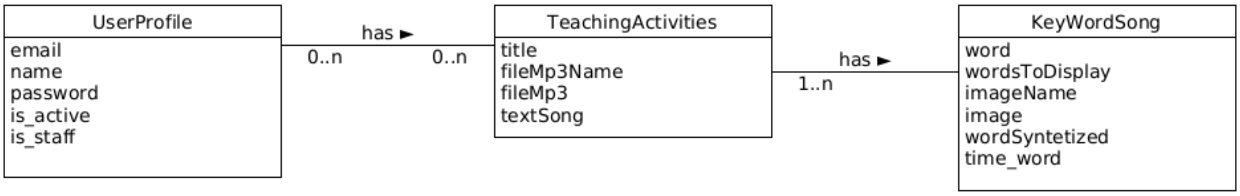


Figura 7: schema ER del db

dove UserProfile identifica gli utenti del sistema. TeachingActivities identifica i singoli brani da cantare e KeyWordSong è la parte della canzone che si sta cantando.

Installing magenta on DJANGO Project

Bisogna lanciare i seguenti comandi:

```
sudo apt-get install build-essential libasound2-dev libjack-dev portaudio19-dev ffmpeg

pip install -r requirements.txt
```

Studio di React

Per prima cosa è necessario installare nvm con questo comando:

```
wget -qO- https://raw.githubusercontent.com/nvm-sh/nvm/v0.40.1/install.sh | bash
```

Per poi installare node con il comando dopo aver riaperto il terminale o riletto il bashrc:

```
nvm install node
```

Infine si installa chakra ui con i seguenti comandi all’interno del progetto react precedentemente creato:

```
npm i @chakra-ui/react @emotion/react


npx @chakra-ui/cli snippet add
```

Per sviluppare il front-end uso visual studio code mentre per il back-end Py-Charm Community Edition.

Chakra UI è una libreria di componenti React per costruire interfacce utente moderne e accessibili. È progettata per semplificare la creazione di applicazioni reattive e facilmente personalizzabili, promuovendo una buona esperienza utente e una gestione semplificata dello stato visivo (come colori, margini, tipografia e altri stili). Chakra UI si concentra molto sull'accessibilità, sulla facilità d'uso e sulla consistenza nell’aspetto visivo, fornendo componenti pronti all'uso che possono essere facilmente personalizzati attraverso una configurazione centralizzata.

Uno dei punti distintivi di Chakra UI è il suo impegno per l'accessibilità. I componenti sono costruiti in modo che siano accessibili per gli utenti con disabilità (ad esempio, utilizzando correttamente gli attributi ARIA, gestendo lo stato visivo per la navigazione tramite tastiera, ecc.).

Chakra UI è progettato per essere **mobile-first** e reattivo. I componenti sono progettati in modo tale da adattarsi automaticamente a diverse dimensioni dello schermo. Puoi anche configurare il comportamento dei componenti a seconda della larghezza del dispositivo (ad esempio, mostrando un layout a colonna per schermi più piccoli e un layout a griglia per schermi più grandi).

L’utilizzo di questo framework semplifica e standardizza di molto il design dell’applicazione così da evitare errori html. Cercherò di usare solo componenti di Chakra UI così non avrò bisogno di validare l’elaborato con strumenti come react-a11y.

Quando si sviluppano applicazioni web moderne, la gestione delle chiamate HTTP al backend è una parte fondamentale. **Axios** è una delle librerie più popolari per effettuare queste richieste, ma ci sono anche altre alternative ampiamente utilizzate, come la **Fetch API** nativa di JavaScript, **jQuery AJAX** e **SuperAgent**. Ognuna di queste librerie ha vantaggi e svantaggi, e la scelta della più adatta dipende dalle specifiche necessità del progetto.

La scelta tra **Axios** e le sue alternative dipende principalmente dalle esigenze specifiche del progetto:

- **Axios** è una soluzione moderna, potente e facile da usare, ideale per applicazioni che richiedono funzionalità avanzate come gli intercettori, la gestione automatica dei dati JSON e una buona gestione degli errori.
- **Fetch API** è una scelta nativa del browser, leggera e adatta per applicazioni semplici, ma con una sintassi più complessa per la gestione degli errori e senza alcune funzionalità avanzate di Axios.
- **jQuery AJAX** è una buona scelta per progetti legacy che già utilizzano jQuery, ma è meno adatta per applicazioni moderne.
- **SuperAgent** è una libreria valida ma meno popolare, che offre alcune funzionalità avanzate per la gestione delle richieste, ma senza il supporto per intercettori e una gestione automatica dei dati JSON.

In generale, **Axios** è la scelta più completa per la maggior parte dei progetti moderni grazie alla sua semplicità d'uso, alla gestione avanzata degli errori e alle funzionalità extra quindi sceglierò questa libreria.

Problematica della voce sulla musica

Nel contesto del mio progetto, ho utilizzato **MusicVAE** di **Magenta**, una rete neurale autoencoder, per la generazione di musica, sfruttando la configurazione **hierdec-trio_16bar**. Successivamente, ho cercato di integrare il cantato, utilizzando un vocoder che avevo sviluppato durante una fase di ricerca preliminare (denominata "spike"). Tuttavia, il risultato ottenuto non era di qualità soddisfacente. In particolare, la qualità del suono vocale era inferiore alle aspettative, quindi ho iniziato a esplorare diverse soluzioni per migliorare il processo.

Dopo aver analizzato approcci professionali di vocalizzazione, ho individuato due potenziali soluzioni al problema:

1. **Creazione di una Rete Neurale MelGAN**: Una possibile soluzione sarebbe l'uso di una rete neurale **MelGAN** (Generative Adversarial Network) per generare la giusta onda sonora da

applicare al file MIDI del cantato. Le reti **MelGAN** sono progettate per estrarre uno spettrogramma dalle informazioni contenute nel MIDI e applicarlo allo strumento vocale, producendo così un suono sintetico della voce. Tuttavia, questo approccio non garantisce un risultato ottimale, poiché la musica è composta da sequenze non continue, con pause tra una parte cantata e l'altra, il che rende difficile ottenere un flusso vocale naturale. La difficoltà principale risiede nel fatto che la musica ha una struttura interrotta, dove i periodi di silenzio tra le note vocali rappresentano una sfida per la modellazione vocale continua.

2. **Uso di Dittonghi Vocali Registrati**: Un altro approccio sarebbe stato quello di creare una serie di **dittonghi vocali**, registrati manualmente e successivamente applicati alle note del MIDI. In questo caso, la frequenza dell'onda sarebbe modificata per adattarsi alla nota specifica, mantenendo però costante il timbro, la forma dell'onda sonora e la sua durata. Sebbene questa tecnica fosse interessante, risultava comunque complessa, in quanto richiedeva la creazione di un ampio catalogo di suoni vocali registrati, nonché la gestione delle variazioni di pitch e timing in modo accurato. Un esempio di questo approccio è openutau, un software open-source progettato per la creazione di musica vocale sintetica. In particolare, è uno strumento utilizzato per generare tracce vocali sintetizzate, molto utilizzato nella comunità di musicisti che producono musica elettronica o musica vocale automatizzata. È compatibile con vari plugin e strumenti per il miglioramento della qualità audio e l'aggiunta di effetti speciali, tra cui vst. I **VST** sono strumenti e effetti digitali che si aggiungono al software di produzione musicale, estendendo le possibilità creative e migliorando la qualità della musica prodotta.

Nonostante entrambe le soluzioni teoriche sembrassero promettenti, il loro impiego si rivelava particolarmente complesso e richiedeva una considerevole quantità di risorse. Di conseguenza, ho deciso di esplorare alternative pratiche e accessibili, cercando tool musicali che potessero facilitare il processo.

Strumenti Utilizzati

Per le prime due canzoni, ho utilizzato la **versione trial di Synthesizer V**, una piattaforma di sintesi vocale basata su intelligenza artificiale. Ho registrato l'output vocale generato da Synthesizer V tramite **Audacity**, un software di registrazione e editing audio. Successivamente, ho combinato la parte vocale con la musica utilizzando uno **script Python** che mi ha permesso di allineare e mixare le tracce in modo efficiente.

Per le successive tre canzoni, ho deciso di cambiare approccio. Invece di utilizzare un sintetizzatore vocale, ho registrato la voce della mia fidanzata, che ha cantato la melodia principale del MIDI per

garantire che la tonalità fosse corretta e naturale. Questo approccio ha permesso di ottenere una base vocale realistica, che poi è stata combinata con voci professionali ottenute dal sito **Kits.ai**. Kits.ai offre servizi vocali avanzati tramite un'interfaccia REST, che consente di integrare le voci sintetiche direttamente in un'applicazione software. Ho scelto le voci che si adattavano meglio al contesto musicale, prestando particolare attenzione all'aderenza al **pitch** originale della melodia.

Poiché non volevo sostenere costi aggiuntivi per un tool professionale specifico, ho preferito registrare la voce generata da Kits.ai tramite Audacity e poi combinare le tracce utilizzando lo stesso script Python impiegato per le prime due canzoni. Questo mi ha permesso di ottenere un buon equilibrio tra qualità vocale e controllo sulla parte musicale, senza dover acquistare strumenti aggiuntivi.

Il processo di generazione del cantato per la musica sintetizzata ha richiesto l'esplorazione di diverse tecniche, tra cui l'uso di reti neurali avanzate e la registrazione di voce umana per migliorare l'accuratezza tonale. Sebbene le soluzioni tecniche avanzate come la rete MelGAN abbiano un potenziale notevole, la combinazione di risorse vocali umane e sintetiche, supportata da tool pratici come Synthesizer V e Kits.ai, ha fornito un buon compromesso tra qualità e praticità, consentendo di superare le sfide iniziali legate alla produzione vocale.

Deploy

Per il deployment dell'applicativo, è stata creata una versione **dockerizzata** dell'applicazione stessa. Successivamente, è stato scelto di utilizzare un ambiente di **deploy gratuito** offerto da Amazon Web Services (AWS), che mette a disposizione una macchina virtuale gratuita per un anno, con le seguenti caratteristiche:

- **1 vCPU**, equivalente a un core di CPU fisico in un'architettura multi-core. Il numero effettivo di core dipende dal tipo di processore fisico utilizzato nei server di Amazon.
- **1 GB di memoria RAM**.
- **1 indirizzo IP pubblico**.
- **Fino a 30 GB di spazio su disco SSD**.

La macchina virtuale non cambia indirizzo IP a meno che non venga spenta e riaccesa, nel qual caso viene assegnata una nuova macchina, ma i file rimangono invariati sulla precedente. Per l'accesso alla macchina, è possibile utilizzare **SSH** oppure configurare il gruppo di sicurezza per aprire le porte desiderate.

Inoltre, per facilitare l'accesso alla macchina, è stato utilizzato il servizio **NoIP** per assegnare un indirizzo **DNS dinamico** alla macchina. Questo permette di evitare il problema di un IP pubblico

che potrebbe cambiare, rendendo l'accesso più stabile e facilmente configurabile. Il sito deployato è presente al seguente indirizzo: **https://karaoke4all.ddns.net/**

Inizialmente, è stato tentato il trasferimento dell'immagine **Docker** sulla macchina, ma il processo non si è rivelato particolarmente agevole. Pertanto, è stato preferito un approccio alternativo, trasferendo prima i file tramite **SCP** e successivamente clonando i sorgenti del progetto sulla macchina.

Un ulteriore ostacolo è stato riscontrato durante la fase di **build** del progetto, in quanto l'immagine Docker necessitava dell'uso di **Conda**, il quale richiede più di 1 GB di RAM per completare la compilazione tramite lo script Docker.

In seguito a queste difficoltà, è stata presa la decisione di semplificare l'architettura dell'applicazione, evitando l'uso di Docker e optando per l'installazione diretta del software sulla macchina virtuale.

Lo schema architetturale della soluzione finale risulta pertanto il seguente:

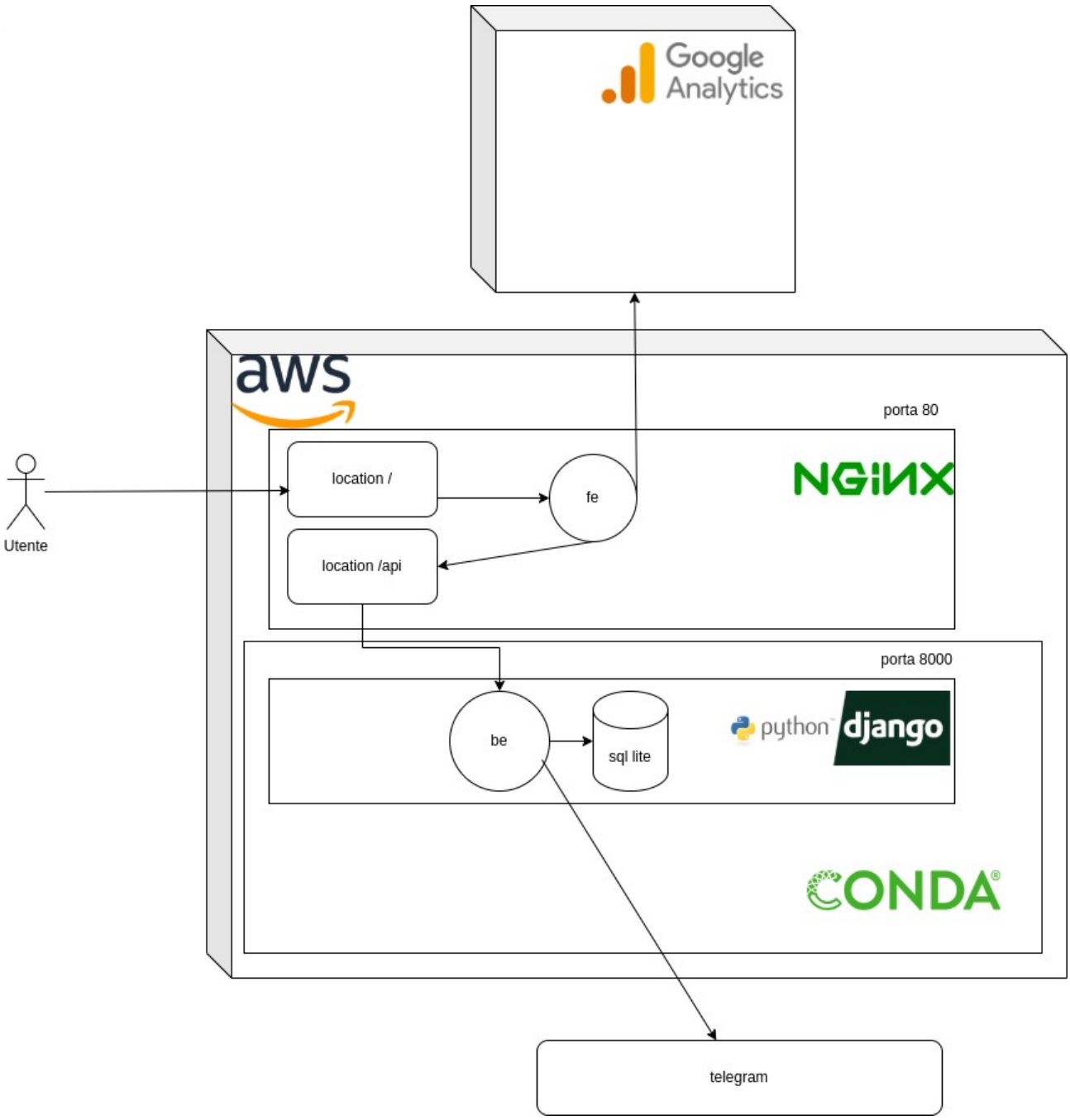


Figura 8: architettura

Come illustrato nell'immagine, la configurazione di **Nginx** prevede due **location**, di cui una funge da **reverse proxy** verso l'applicazione **Django**, pubblicata sulla porta **8000**. Quest'ultima è connessa a un **database interno** basato su file.

L'uso del reverse proxy è stato adottato per evitare problematiche legate al **cross-origin resource sharing (CORS)** e ai conflitti di **cross-location**, garantendo una gestione centralizzata delle richieste HTTP tra il client e il server.

Il sito è stato esposto utilizzando un certificato SSL/TLS emesso da **Let's Encrypt**, ottenuto tramite il tool **Certbot**. Questo certificato è stato configurato per garantire la sicurezza delle comunicazioni tra il client e il server, criptando il traffico e garantendo l'autenticità del sito. Let's Encrypt fornisce certificati SSL/TLS gratuiti e automatizzati, consentendo di implementare facilmente una connessione sicura HTTPS per il sito.

Il **frontend** dell'applicazione raccoglie i dati relativi alle visite delle singole pagine e agli eventi di interazione con il contenuto, come il **play** delle canzoni, utilizzando **Google Analytics**, ma questa raccolta avviene esclusivamente nell'ambiente di **produzione**. In particolare, vengono tracciate le visite alle pagine del sito e gli eventi specifici legati all'interazione dell'utente con le tracce audio. Questi dati vengono inviati a **Google Analytics** solo nell'ambiente di produzione, garantendo che le informazioni raccolte siano pertinenti e precise, mentre nell'ambiente di sviluppo la raccolta dei dati è disabilitata. Questo approccio consente di evitare l'inclusione di dati di test o di sviluppo nei report di analisi, assicurando una visione accurata del comportamento degli utenti reali nell'ambiente di produzione.

Questa una schermata del report di google analytics:

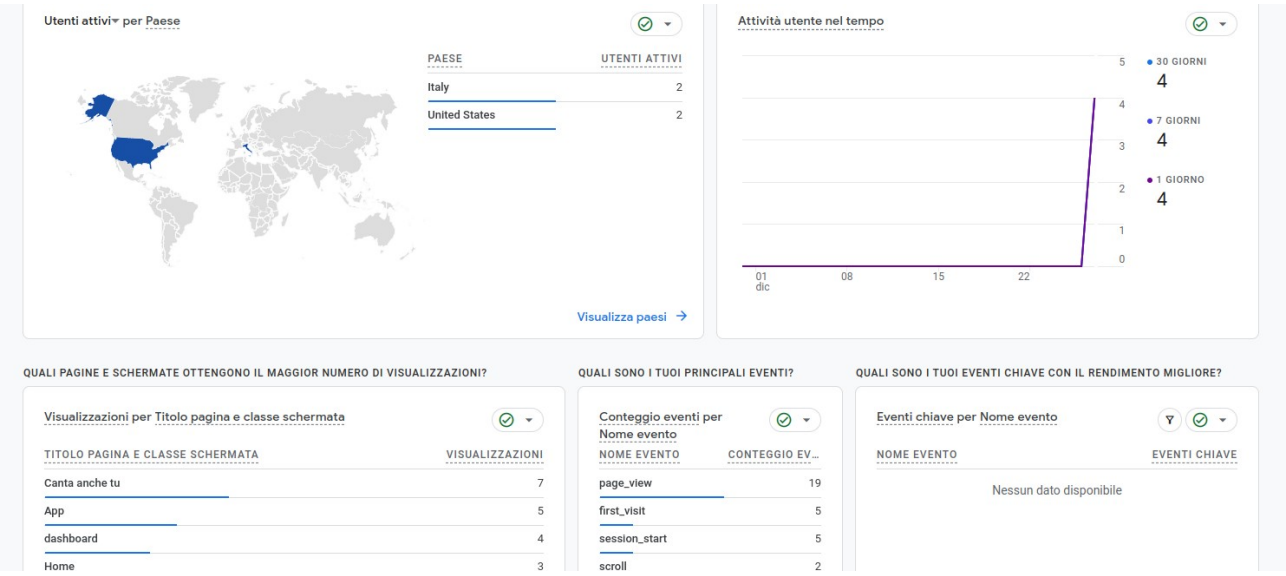


Figura 10: google analitycs

Questo report riguarda un sito privo di attività pubblicitaria o di marketing. Gli utenti potenziali del sito sono principalmente amici, parenti e il professore. Al momento, il sito non è stato ancora condiviso con alcun ente per scopi di test.

L'applicazione Django è eseguita all'interno di un ambiente **Conda**, che consente di gestire e isolare le dipendenze del progetto, in particolare per quanto riguarda la versione di Python. In questo caso, il sistema operativo utilizza Python 3.12 come versione di sistema, mentre Django è configurato per utilizzare Python 3.8. Questo approccio garantisce che l'ambiente di esecuzione di Django sia compatibile con le librerie e i pacchetti richiesti, senza interferire con le configurazioni globali del sistema.

La macchina è stata preparata con i seguenti comandi:

comandi	spiegazione
wget https://repo.anaconda.com/miniconda/Miniconda3-latest-Linux-x86_64.sh chmod +x Miniconda3-latest-Linux-x86_64.sh ./Miniconda3-latest-Linux-x86_64.sh	Installazione di conda
sudo apt-get update && sudo apt-get install -y \ fluid-soundfont-gm \ build-essential \ libasound2-dev \ libjack-dev \ ffmpeg \ nodejs \ npm \ sox	Installazione pacchetti utili per l'applicazione
conda create --name ai_project python=3.8 conda init source ~/.bashrc conda activate ai_project conda install pip pip install -r tesi-main/karaoke_app_be/requirements.txt	Installazione di environment in 2 step così da far bastare 1 GB di ram
sudo apt install nginx	Installazione di nginx
npm install npm run build	Creazione del pacchetto di frontend per produzione

modifico root /etc/nginx/sites-available/default di modo che punta a dist sudo chmod 755 /home/ubuntu sudo chmod 755 /home/ubuntu/tesi-main sudo chmod 755 /home/ubuntu/tesi-main/karaoke_app_fe sudo chown -R www-data:www-data /home/ubuntu/tesi-main/karaoke_app_fe/dist/ sudo chmod -R 755 /home/ubuntu/tesi-main/karaoke_app_fe/dist/ sudo systemctl restart nginx	Faccio puntare nginx al pacchetto compilato e rendo pubblico il path del pacchetto. Infine riavvio il server
sudo apt install certbot python3-certbot-nginx sudo certbot --nginx	Creo un certificato con il bot di letsencrypt

La configurazione nginx risultante sarà quindi questa:

```
server {
        # SSL configuration
        #
        listen 443 ssl default_server;
        listen [::]:443 ssl default_server;
        ssl_certificate /etc/letsencrypt/live/karaoke4all.ddns.net/fullchain.pem; # managed by Certbot
    ssl_certificate_key /etc/letsencrypt/live/karaoke4all.ddns.net/privkey.pem; # managed by Certbot
    include /etc/letsencrypt/options-ssl-nginx.conf; # managed by Certbot
    ssl_dhparam /etc/letsencrypt/ssl-dhparams.pem;


            # Configurazione SSL di base
            #ssl_protocols TLSv1.2 TLSv1.3;
            #ssl_ciphers 'ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-RSA-AES128-GCM-
SHA256';
```

```
    root /home/ubuntu/tesi-main/karaoke_app_fe/dist;

    # Add index.php to the list if you are using PHP
    index index.html index.htm index.nginx-debian.html;

    server_name _;

    location / {
        # First attempt to serve request as file, then
        # as directory, then fall back to displaying a 404.
        try_files $uri $uri/ /index.html;
    }

    location /api {
                proxy_pass http://127.0.0.1:8000/api;  # Gunicorn o Django development server
                proxy_set_header Host $host;
                proxy_set_header X-Real-IP $remote_addr;
                proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
                proxy_set_header X-Forwarded-Proto $scheme;
       }
}
server {
        listen 80;
        server_name karaoke4all.ddns.net;
```

```
        return 301 https://$host$request_uri;

}
```

La configurazione del server Nginx presentata gestisce il traffico HTTPS e HTTP per il sito web, utilizzando un certificato SSL/TLS emesso da **Let's Encrypt**. La configurazione è suddivisa in due blocchi di server: uno per il traffico sicuro tramite il protocollo HTTPS (porta 443) e uno per il traffico HTTP (porta 80), con un redirect verso HTTPS.

Il primo blocco di configurazione è destinato al traffico **HTTPS**, che ascolta sulla porta 443. La configurazione include le seguenti direttive:

- `listen 443 ssl`: Nginx è configurato per ascoltare sulla porta 443, abilitando la modalità SSL per garantire una connessione sicura.
- `ssl_certificate` **e** `ssl_certificate_key`: Queste direttive specificano i percorsi del certificato SSL e della chiave privata generati e gestiti tramite **Certbot**, uno strumento automatizzato per ottenere certificati SSL da **Let's Encrypt**. Il certificato utilizzato è `fullchain.pem` e la chiave è `privkey.pem`.
- `include /etc/letsencrypt/options-ssl-nginx.conf`: Includendo il file di configurazione predefinito di **Let's Encrypt**, Nginx applica le migliori pratiche di configurazione SSL per garantire una comunicazione sicura.
- `ssl_dhparam /etc/letsencrypt/ssl-dhparams.pem`: Definisce i parametri di Diffie-Hellman per una maggiore sicurezza nelle negoziazioni SSL/TLS.

In questa sezione è definito anche il **document root**, che punta alla directory contenente i file statici del front-end dell'applicazione: `/home/ubuntu/tesi-main/karaoke_app_fe/dist`. Nginx è configurato per servire questi file statici come la pagina predefinita del sito, con la direttiva `index` che definisce i file di indice da cercare (ad esempio, `index.html`).

La sezione `location /` gestisce le richieste verso la root del sito e utilizza la direttiva `try_files` per cercare prima un file corrispondente, poi una directory e, infine, restituisce il file `index.html` nel caso in cui non venga trovato nulla. Questo approccio è tipico nelle applicazioni web single-page (SPA), dove tutte le rotte sono gestite dal JavaScript lato client.

La sezione `location /api` definisce un **reverse proxy** che instrada le richieste API alla porta 8000, dove il backend **Django** (o un server Gunicorn) è in esecuzione. Le direttive `proxy_set_header` configurano correttamente gli header HTTP per il proxy, garantendo che le informazioni sul client (come l'indirizzo IP remoto e il protocollo utilizzato) vengano trasmesse correttamente al backend.

Il secondo blocco di configurazione gestisce le richieste in ingresso sulla porta **80** (HTTP). Quando un client tenta di connettersi tramite HTTP, la direttiva `return 301 https://$host$request_uri;` forza il redirect permanente delle richieste verso il protocollo sicuro HTTPS, utilizzando il **codice di stato HTTP 301** per indicare un redirect permanente. Questo approccio garantisce che tutte le comunicazioni avvengano su una connessione sicura.

Addestramento modello magenta

Per l'addestramento del modello, ho scelto di utilizzare Google Colab, una piattaforma che offre l'accesso gratuito a risorse computazionali avanzate, come le GPU e le TPU. In particolare, ho utilizzato la **GPU Tesla T4**, che è una delle GPU più performanti offerte da Google Colab per il calcolo intensivo, come l'addestramento di modelli di machine learning.

La **GPU Tesla T4** è una scheda grafica basata sull'architettura Turing di NVIDIA. Essa offre un'ottima combinazione di prestazioni e efficienza energetica, con una memoria video di 16 GB GDDR6 e capacità elevate di parallelizzazione, che la rendono particolarmente adatta per carichi di lavoro di deep learning e training di modelli complessi come quelli basati su reti neurali. Le sue caratteristiche la rendono una scelta ideale per applicazioni che richiedono calcoli in parallelo e una gestione ottimizzata di grandi dataset.

Anche se le **TPU (Tensor Processing Units)** sono generalmente più performanti delle GPU per determinati tipi di operazioni legate al machine learning, in particolare per reti neurali di grandi dimensioni, l'accesso alle TPU su Google Colab è soggetto a limiti e risorse a pagamento. Le TPU sono progettate specificamente per accelerare il calcolo delle operazioni tensoriali, ma a differenza delle GPU, che possono essere utilizzate in modo più flessibile, le TPU sono ottimizzate per determinate applicazioni di deep learning e potrebbero non essere ideali per tutti i tipi di carico di lavoro. Inoltre, l'uso continuativo delle TPU è disponibile solo a pagamento, mentre le GPU come la Tesla T4 sono gratuite per periodi limitati, rendendo quest'ultima una scelta pratica per gli utenti con budget limitati.

Ho deciso di installare l'ambiente locale utilizzando **Conda**, proprio come avrei fatto su una macchina locale, ma ho dovuto adattare alcune operazioni per sfruttare il sistema in Google Colab. Infatti, una limitazione importante di Colab è che non è possibile attivare direttamente un ambiente Conda utilizzando i comandi tradizionali, come avverrebbe in un ambiente locale o su un server dedicato. In Colab, l'esecuzione dei comandi deve avvenire attraverso l'interfaccia di shell, che non consente l'attivazione di ambienti Conda in modo nativo.

Per superare questa difficoltà, ho dovuto utilizzare comandi specifici per **eseguire operazioni all'interno di un ambiente Conda** già esistente, sfruttando l'istruzione `!conda run -n py38`, dove `py38` rappresenta il nome dell'ambiente Conda da utilizzare. Un esempio di tale operazione è il comando:

```
!conda run -n py38 pip uninstall -y numpy
```

Questo comando permette di eseguire l'operazione di disinstallazione del pacchetto `numpy` all'interno dell'ambiente Conda `py38`, senza la necessità di attivarlo direttamente. L'uso di `!conda run` permette quindi di eseguire operazioni in un ambiente Conda in modo trasparente, simulando l'attivazione dell'ambiente e mantenendo la compatibilità con il sistema di Colab.

Questa soluzione mi ha permesso di mantenere la configurazione del mio ambiente di sviluppo coerente con quella locale, pur utilizzando la piattaforma di Google Colab per beneficiare della potenza delle GPU e delle risorse computazionali messe a disposizione.

Inoltre, per gestire e conservare i dati e i modelli addestrati, ho montato **Google Drive** all'interno di Google Colab, utilizzandolo come repository per i file necessari all'esecuzione del progetto. Il montaggio di Google Drive permette di accedere direttamente ai propri file su Drive come se fossero parte del file system di Colab, facilitando il salvataggio e il recupero dei dati durante l'addestramento.

Ho scelto di utilizzare Google Drive per vari motivi. In primo luogo, offre una grande capacità di archiviazione (fino a 50 GB gratuitamente, con possibilità di espandere lo spazio tramite piani a pagamento), che è ideale per gestire dataset di grandi dimensioni e modelli complessi. Inoltre, Google Drive garantisce una sincronizzazione continua dei file tra i vari dispositivi, permettendo di lavorare su diversi dispositivi senza preoccuparsi della gestione fisica dei dati.

Durante il progetto, ho utilizzato Drive principalmente per archiviare i **dataset di addestramento**, i **modelli salvati** e i **log di esecuzione**. Questo approccio mi ha consentito di avere una copia sicura dei dati, di monitorare l'avanzamento dell'addestramento in tempo reale e di eseguire operazioni come il salvataggio dei modelli intermedi durante il processo di training.

Per poter sfruttare appieno la **GPU** su Google Colab, ho dovuto installare i driver necessari, in particolare quelli relativi a **CUDA** e **cuDNN**, che sono essenziali per accelerare i calcoli durante l'addestramento del modello di deep learning. In Colab, anche se le GPU sono disponibili, è necessario configurare correttamente l'ambiente per poterle utilizzare in modo efficiente.

I due comandi che ho utilizzato sono i seguenti:

```
!sudo apt install nvidia-cuda-toolkit=11.5.1-1ubuntu1
```

Questo comando installa il **CUDA Toolkit**, una raccolta di strumenti e librerie necessarie per sviluppare applicazioni che utilizzano GPU NVIDIA. In particolare, la versione `11.5.1-1ubuntu1` è compatibile con l'ambiente di Colab e con la versione della GPU T4 che ho utilizzato. CUDA (Compute Unified Device Architecture) è essenziale per eseguire operazioni parallele sulle GPU, accelerando così i processi di addestramento e inferenza per i modelli di machine learning. L'installazione di questo toolkit consente di utilizzare le funzioni di calcolo avanzato sulle GPU senza dover scrivere codice specifico per gestire manualmente le operazioni parallele.

Il secondo comando è:

```
!apt-get install -y libcudnn8 libcudnn8-dev
```

Il comando sopra installa **cuDNN** (CUDA Deep Neural Network library), una libreria fondamentale per l'accelerazione delle operazioni di deep learning, ottimizzata per le GPU NVIDIA. In particolare:

- `libcudnn8` è la libreria principale di cuDNN che fornisce funzioni per operazioni come convoluzioni, normalizzazione, pooling, e altre operazioni comuni nelle reti neurali.
- `libcudnn8-dev` è il pacchetto di sviluppo che include i file necessari per compilare applicazioni che utilizzano cuDNN.

L'installazione di cuDNN è fondamentale per sfruttare al massimo la potenza di calcolo delle GPU, in quanto ottimizza le operazioni di addestramento di reti neurali complesse, garantendo prestazioni superiori rispetto all'esecuzione su CPU.

Questi driver e librerie sono cruciali per sfruttare le capacità hardware delle GPU in modo efficiente e accelerare i tempi di addestramento dei modelli. Dopo aver completato l'installazione, sono stato in grado di utilizzare la GPU T4 su Colab per eseguire il training del modello con tempi di esecuzione significativamente ridotti rispetto all'uso della sola CPU.

Ho lanciato il comando per avviare il processo di **training** del modello MusicVAE, una rete neurale per la generazione di musica. Il comando completo che ho utilizzato è il seguente:

```
!conda run -n py38 music_vae_train \

 --config=hierdec-trio_16bar \
```

```
--run_dir="/content/drive/MyDrive/training" \
--mode=train \
--examples_path=$SEQUENCES_TFRECORD \
--hparams=batch_size=16,learning_rate=0.0005 \
--checkpoints_to_keep=5
```

L'opzione `config` definisce la configurazione del modello da utilizzare. In questo caso, `hierdec-trio_16bar` è una configurazione specifica per addestrare un modello VAE su sequenze musicali di 16 barre (16 bar). La configurazione `hierdec-trio` si riferisce a un tipo di VAE gerarchico, ottimizzato per modelli musicali complessi e strutturati.

L'opzione run_dir specifica la directory in cui verranno salvati i risultati del training, inclusi i checkpoint del modello e i log di esecuzione. In questo caso, la directory si trova su **Google Drive**, per consentire una facile gestione dei file e un accesso persistente anche dopo la sessione su Google Colab.

L'opzione `mode` indica che il comando deve eseguire il **training** del modello. Esistono altre modalità come `generate` o `evaluate`, ma in questo caso l'obiettivo è addestrare il modello.

Il parametro example_path definisce il percorso degli **esempi di dati** per l'addestramento. `SEQUENCES_TFRECORD` è una variabile d'ambiente che fa riferimento al file contenente il dataset in formato TFRecord. Questo formato è ampiamente utilizzato in TensorFlow per l'efficienza nella gestione di grandi volumi di dati.

L'opzione `hparams` consente di definire **iperparametri** personalizzati per il training. In questo caso:

- `batch_size=16`: definisce la dimensione del batch durante l'addestramento, ovvero il numero di esempi che vengono processati in ogni passo del modello prima di aggiornare i pesi. Una dimensione di batch di 16 è una scelta comune che bilancia efficienza computazionale e stabilità.
- `learning_rate=0.0005`: imposta il tasso di apprendimento, che controlla la velocità con cui i pesi del modello vengono aggiornati durante l'addestramento. Un valore di `0.0005` è piuttosto piccolo, il che aiuta a evitare aggiornamenti troppo rapidi che potrebbero compromettere la convergenza del modello.

L'opzione checkpoints_to_keep specifica il numero massimo di **checkpoint** del modello che devono essere conservati durante l'addestramento. I checkpoint sono versioni intermedie del

modello salvate durante il processo di addestramento. Mantenere solo gli ultimi 5 checkpoint consente di risparmiare spazio su disco, pur avendo un numero sufficiente di salvataggi per monitorare il progresso e recuperare in caso di errori.

Per generare musica, ho utilizzato un **checkpoint specifico** ottenuto durante l'addestramento del modello. In particolare, il comando seguente è stato utilizzato per caricare il checkpoint e avviare il processo di generazione musicale.

Durante il processo di generazione della musica, uno degli ostacoli che ho dovuto affrontare è stato l'individuazione del **nome corretto del checkpoint** da utilizzare. In particolare, i file di checkpoint salvati durante l'addestramento avevano un'estensione, come ad esempio `.index` o `.data`, mentre il parametro da passare nel comando per la generazione della musica richiede solo il **nome del file senza estensione**.

I checkpoint salvati nel formato TensorFlow, infatti, sono composti da più file, ognuno con un'estensione specifica:

- `checkpoint_100000.index`: contiene l'indice del checkpoint.
- `checkpoint_100000.data`: contiene i dati del checkpoint.
- `checkpoint_100000.meta`: contiene le informazioni del modello.

Tuttavia, nel comando per la generazione della musica, il parametro `--checkpoint_file` richiede solo il **nome base** del checkpoint, **senza estensione**. Ad esempio, se il file salvato si chiama `checkpoint_100000.index`, il parametro deve essere passato come:

```
--checkpoint_file="checkpoint_100000"
```

Durante il processo di addestramento e generazione della musica, sono emersi alcuni **problemi tecnici** relativi alle risorse di sistema e alla gestione dello spazio di archiviazione. Questi problemi sono stati affrontati con diverse soluzioni pratiche.

1. Problema con il batch_size troppo elevato

Un primo problema si è presentato quando ho tentato di aumentare il valore del **batch_size** durante l'addestramento. Inizialmente, ho impostato un valore di `batch_size=32`, ma la GPU non è riuscita a gestire il carico di lavoro. Il processo ha causato un **esaurimento della memoria GPU**, portando al fallimento del training. Questo è un problema comune quando si lavora con modelli complessi e risorse limitate, poiché un batch size troppo grande può richiedere più memoria di quanto la GPU possa supportare.

Soluzione:

Per risolvere questo problema, ho ridotto il valore del **batch_size** a 16, che ha permesso di mantenere l'addestramento senza sovraccaricare la memoria della GPU. Ridurre il batch size implica elaborare più piccoli gruppi di dati per volta, ma ha permesso di ottimizzare l'uso delle risorse disponibili senza compromettere troppo le performance del modello.

Questa è un'immagine delle risorse utilizzate durante l'addestramento ed il primo picco è dove l'elaborazione è andata in errore.

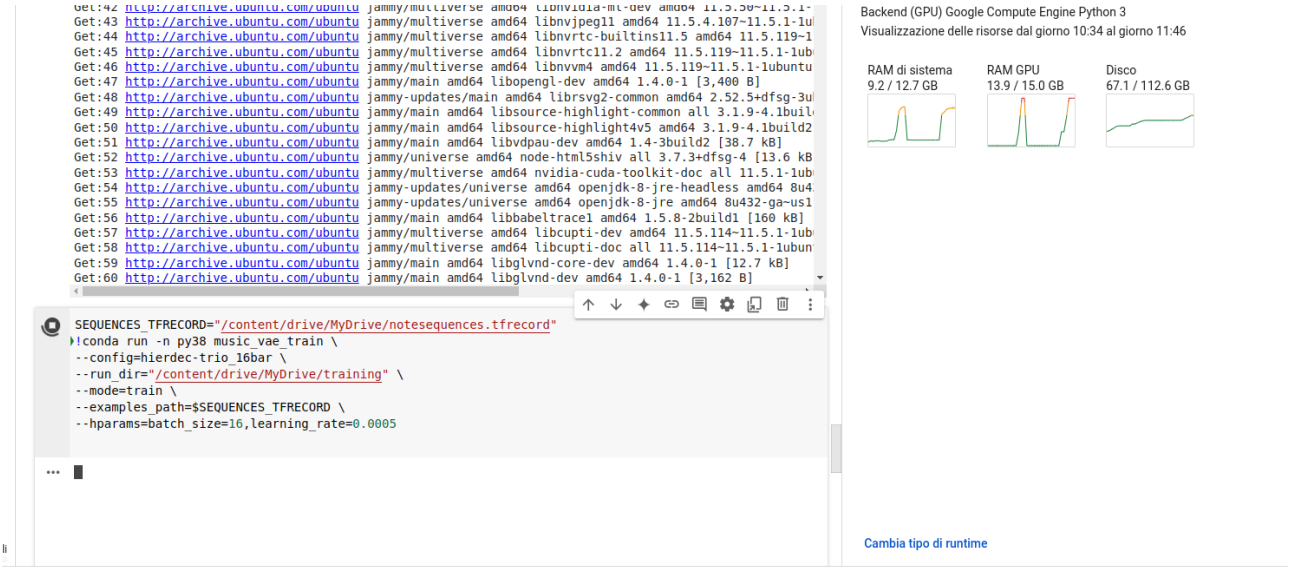


Figura 11: google colab, training

2. Problema di spazio insufficiente

Un altro problema che si è presentato è stato il **problema di spazio insufficiente** su Google Drive, dove venivano salvati i dati di addestramento, i checkpoint del modello e le sequenze musicali generate. Con l'accumulo continuo di file, lo spazio su Google Drive è diventato limitato, rischiando di bloccare ulteriormente il processo di addestramento e generazione.

Soluzione:

Per affrontare questo problema di spazio, ho implementato uno **script su Google Apps Script** che esegue automaticamente la **pulizia del cestino di Google Drive** a intervalli regolari. Lo script è stato configurato per eliminare i file nel cestino ogni **1 minuto**, liberando rapidamente spazio su Google Drive e evitando che il sistema si bloccasse a causa di spazio insufficiente. Questo approccio ha permesso di mantenere il flusso di lavoro ininterrotto, senza necessità di intervenire manualmente.

Lo script di Google Apps Script è stato impostato per eseguire una funzione di **eliminazione automatica dei file nel cestino**, riducendo così il rischio di esaurire lo spazio di archiviazione.

Questo lo script:

```
function svuotaCestino() {
  var files = DriveApp.getTrashedFiles(); // Ottieni i file nel Cestino
  while (files.hasNext()) {
    var file = files.next();
    var fileId = file.getId();

    // Usa Google Drive API per eliminare definitivamente il file
    var url = 'https://www.googleapis.com/drive/v3/files/' + fileId;

    var options = {
      'method': 'delete',
      'muteHttpExceptions': true,
      'headers': {
        'Authorization': 'Bearer ' + ScriptApp.getOAuthToken()
      }
    };

    // Effettua la richiesta per eliminare definitivamente il file
    var response = UrlFetchApp.fetch(url, options);
    if (response.getResponseCode() === 204) {
      Logger.log('File eliminato definitivamente: ' + file.getName());
    } else {
      Logger.log('Errore nell\'eliminazione del file: ' + file.getName());
    }
  }
```

```
}
```

3. Disconnessione dal server colab

Un altro problema significativo che ho incontrato durante l'addestramento e la generazione della musica è stato il rischio di disconnessione dal server di Google Colab. Questo tipo di disconnessione è un inconveniente comune durante l'utilizzo di Google Colab, specialmente per i processi di lunga durata come l'addestramento di modelli complessi. Colab tende a disconnettersi automaticamente se non c'è attività nel notebook per un certo periodo di tempo, o se il processo di addestramento impiega troppe risorse per un periodo prolungato.

Le disconnessioni frequenti possono causare la perdita di avanzamenti nel training, con conseguente necessità di riavviare l'intero processo, il che rappresenta una notevole perdita di tempo e risorse.

Per superare temporaneamente questo problema, ho deciso di accontentarmi dei checkpoint generati dal modello fino al momento della disconnessione. Sebbene la disconnessione interrompesse il processo di addestramento, i checkpoint creati durante il processo di training erano comunque utilizzabili per continuare con la generazione musicale.

In un primo momento, ho utilizzato i checkpoint salvati automaticamente da Colab fino al punto di disconnessione. Nonostante l'interruzione, questi checkpoint contenevano informazioni sufficienti sullo stato del modello per generare musica di qualità accettabile, sebbene il modello non fosse stato addestrato fino al suo pieno potenziale.

I tentativi sono stati molteplici:

Learning rate	**Batch size**	**Risultato**
0.0005	16	Checkpoint non utilizzabile
0.001	16	Checkpoint forse leggermente migliore del precedente
0.005	16	GPU exception
0.001	32	GPU exception

Valutazione

Per la valutazione, ho coinvolto due associazioni tramite un questionario e ho richiesto un feedback a tre professori. Le associazioni contattate sono la **Fondazione Time 2** (https://fondazionetime2.it/) e il **Progetto Ponte IL Margine**.

La **Fondazione Time 2** ha evidenziato alcune criticità riguardo alla presentazione del progetto online. In particolare, ha segnalato che la homepage non fornisce una spiegazione chiara e

immediata del progetto, rendendo difficile per l'utente comprendere il contesto e gli obiettivi. Inoltre, sono state criticate le immagini, giudicate troppo complesse rispetto ai testi che presentano concetti poco concreti. La Fondazione ha quindi espresso una valutazione negativa non tanto sulla qualità musicale delle canzoni, quanto sull'efficacia comunicativa del materiale visivo e testuale, che non risulta sufficientemente chiaro e accessibile per il pubblico di riferimento.

Per rispondere a queste critiche costruttive, ho deciso di migliorare la homepage e aggiungere una nuova canzone, accessibile esclusivamente agli utenti loggati. Si tratta di una salsa, pensata per essere non solo piacevole da ascoltare, ma anche divertente da ballare. La canzone è stata realizzata utilizzando il servizio di Suno.ai, il che garantisce una qualità professionale, ed è stata poi modificata per adattarsi alla lunghezza ideale per un esercizio di canto.

Inoltre, ho implementato un nuovo form nella pagina dei contatti, tramite il quale le richieste di nuove canzoni vengono inviate direttamente a un bot su Telegram.

Nella fase di apprendimento, è possibile anche registrare un video del ragazzo mentre sperimenta il canto della canzone. Questo permette non solo di avere un ricordo tangibile del momento, ma anche di analizzare e studiare eventuali errori commessi durante l’esecuzione. La registrazione video fornisce uno strumento utile per osservare le tecniche vocali, i movimenti e la postura, facilitando l'individuazione di aree da migliorare e promuovendo un apprendimento più consapevole e mirato.

Durante lo sviluppo del gioco educativo, è emerso un feedback importante da parte del Progetto Ponte: il tempo a disposizione per ripetere correttamente una parola durante le varie fasi del gioco risultava insufficiente, in particolare per i partecipanti con difficoltà nel pronunciare le parole o con tempi di risposta più lunghi. Questo aspetto veniva segnalato come un ostacolo nell’esperienza di gioco, creando un senso di frustrazione e diminuendo l’efficacia dell’apprendimento.

Per ovviare a questa problematica, sono state introdotte due modifiche principali al funzionamento del gioco:

1. **Aumento del Tempo di Ripetizione**
   Per garantire che ogni utente avesse un tempo sufficiente per ripetere la parola, è stato deciso di aggiungere un intervallo extra di 2 secondi alla durata dell’audio e alla visualizzazione della parola stessa. In questo modo, i partecipanti hanno a disposizione più tempo per ascoltare e ripetere la parola prima di passare alla successiva. L'aumento del tempo ha contribuito a ridurre il senso di urgenza e a migliorare l'esperienza complessiva.
2. **Funzionalità "Torna Indietro"**

   Inoltre, per dare agli utenti un maggiore controllo sul processo di ripetizione, è stata implementata una nuova funzionalità: il pulsante **"Torna Indietro"**. Questo pulsante consente agli utenti di ripetere l’ultima parola presentata, permettendo loro di esercitarsi nuovamente senza dover aspettare che il gioco proceda autonomamente. La possibilità di

ripetere una parola è stata una soluzione particolarmente apprezzata, in quanto ha permesso agli utenti di correggere eventuali errori di pronuncia e migliorare la propria performance in tempo reale.

Per raccogliere feedback più ampi e mirati sul progetto e sulle modifiche implementate, sono stati avviati tentativi di contatto con diverse associazioni che si occupano di disabilità e inclusione sociale. In particolare, sono state contattate le seguenti organizzazioni:

- AIPD (Associazione Italiana Persone Down)
- FLI (Federazione logopedisti italiani)
- FISH Onlus (Federazione Italiana per il Superamento dell'Handicap)
- AIRDown (Associazione Italiana Ricerca e Sostegno alle persone con Sindrome di Down)
- ANFFAS (Associazione Nazionale Famiglie di Persone con Disabilità Intellettiva e/o Relazionale)

Nonostante gli sforzi e i tempi di attesa, purtroppo non è stato possibile ricevere feedback formali da parte di queste associazioni entro la scadenza prevista per il completamento di questa fase del progetto. Tuttavia, questo non ha ridotto l'importanza del tentativo di coinvolgere attivamente le organizzazioni, che rimangono un punto di riferimento fondamentale per la validazione e il miglioramento dei progetti di inclusione.

Grazie al professore Ciravegna ho potuto contattare anche l’associazione La Perla. Il feedback positivo ricevuto, che includeva anche il suggerimento di aggiungere brani di facile accesso, mi ha incoraggiato a inserire tra i brani disponibili per il karaoke la canzone "La vecchia fattoria". Questa scelta, oltre a rispondere alla richiesta di canzoni note e coinvolgenti, è stata favorita dal fatto che il brano non richiede diritti d'autore alla SIAE, rendendolo una soluzione vantaggiosa sia dal punto di vista legale che economico.

Conclusioni

A partire dall'analisi dei requisiti e dalla valutazione preliminare, ho sviluppato un prototipo del progetto complessivo. Questo prototipo ha rappresentato una fase cruciale per testare e convalidare la fattibilità dell'idea, permettendo di esplorare le potenzialità e le criticità del sistema in un ambiente controllato. Grazie alla realizzazione del prototipo, è stato possibile ottenere un riscontro concreto sulla funzionalità e sull'efficacia delle soluzioni progettate, nonché raccogliere feedback utili per affinare ulteriormente il progetto. Questo processo ha permesso di identificare eventuali

punti di miglioramento e di ottimizzare la soluzione, garantendo un'elevata qualità del prodotto finale pur non sviluppando tutte le funzionalità ipotizzate.

I requisiti funzionali non completati per intero sono:

- **Generazione di Brani Musicali**

  Se dovessero richiedermi altre canzoni per il sito userei suno.ai per la generazione delle canzoni

- **Riconoscimento Facciale**

  Nel corso dello sviluppo, ho valutato il beneficio derivante dal rapporto tra impatto ed effort per ogni funzionalità proposta. In particolare, ho deciso di posticipare la valutazione automatizzata dell'umore dei ragazzi a una futura release del software. Questa scelta è stata motivata dal fatto che, in una fase iniziale di utilizzo del sistema, i ragazzi potrebbero essere supportati da personale qualificato, il quale potrebbe monitorare e assistere l'utente con disabilità in modo più efficace. L'integrazione di una valutazione automatica dell'umore sarà comunque presa in considerazione in una fase successiva, quando il sistema avrà raggiunto una maggiore maturità e sarà stato utilizzato in modo più autonomo dagli utenti.

I requisiti tecnici inizialmente ipotizzati sono stati adattati alle esigenze di budget, alle specifiche funzionali e ai feedback ricevuti dagli utenti durante il processo di sviluppo. È stata implementata una soluzione dockerizzata per il prodotto, tuttavia, per il suo corretto deployment è necessario acquistare macchine con una maggiore capacità di RAM.

Per quanto riguarda la componente mobile, si è deciso di non svilupparla, poiché è emerso che l'utilizzo di uno schermo di dimensioni ridotte portava l'utente a concentrarsi principalmente sulla musica, tralasciando l'aspetto del canto, che era invece fondamentale per l'esperienza complessiva del sistema.

Nonostante l'assenza di una versione mobile dedicata, le pagine web sono state progettate per essere responsive, garantendo un'ottima fruibilità su dispositivi con diverse dimensioni di schermo. Inoltre, sono state adottate soluzioni per rendere il sistema altamente accessibile, assicurando che l'interfaccia utente sia utilizzabile da persone con diverse abilità e che l'esperienza d'uso sia inclusiva per tutti. Il player non funziona correttamente su tutti i dispositivi mobili. Questo limite è emerso a causa delle restrizioni imposte dai browser sui dispositivi mobili, che talvolta impediscono il corretto funzionamento di alcuni componenti multimediali, come i player audio e la registrazione video, in particolar modo per quanto riguarda l'interazione con funzionalità avanzate. Pertanto, mentre l'interfaccia è ottimizzata per essere visualizzata su schermi di diverse dimensioni, il player

richiede un'ulteriore riflessione per garantire il suo pieno funzionamento anche su dispositivi mobili.

In sintesi, ho sviluppato un prototipo che mi ha permesso di valutare l’efficacia dell’idea e, al contempo, di approfondire l’utilizzo di TensorFlow per la creazione di modelli, l'applicazione di tecniche di fitting con Magenta e l'uso di Python per la gestione e l'elaborazione di brani musicali.

Futuri sviluppi

Se le richieste di creazione di nuove canzoni dovessero diventare troppo numerose, si potrebbe valutare l'opportunità di automatizzare il processo e, eventualmente, sviluppare un modello personalizzato basato su Magenta per la generazione automatica delle canzoni.

In ogni caso, l'ampliamento del numero di tracce disponibili per il prototipo rappresenta una delle principali linee di sviluppo. A tale scopo, intendo arricchire la selezione musicale includendo canzoni note e piacevoli, in grado di attrarre un pubblico più ampio. L'introduzione di brani di grande successo contribuirà a migliorare l'esperienza utente, rendendo il prototipo non solo più vario, ma anche più interessante dal punto di vista emotivo e coinvolgente.

Appendice: intelligenza artificiale

L'IA si può definire come l'insieme di tecnologie che permettono alle macchine di simulare comportamenti intelligenti, come il riconoscimento vocale, la visione artificiale, la comprensione del linguaggio naturale, la presa di decisioni autonome e l'apprendimento da esperienze passate.

L'intelligenza artificiale (IA) è un campo ampio e multidisciplinare che include, tra le sue principali aree di studio e applicazione, il *machine learning*, la *robotica* e la *computer vision*. Questi sotto-campi si interconnettono e contribuiscono a definire la complessità dell'intelligenza artificiale, ciascuno focalizzandosi su aspetti specifici ma complementari dell'automazione e dell'apprendimento. Un possibile schema che rappresenta questa interrelazione potrebbe essere strutturato come segue:

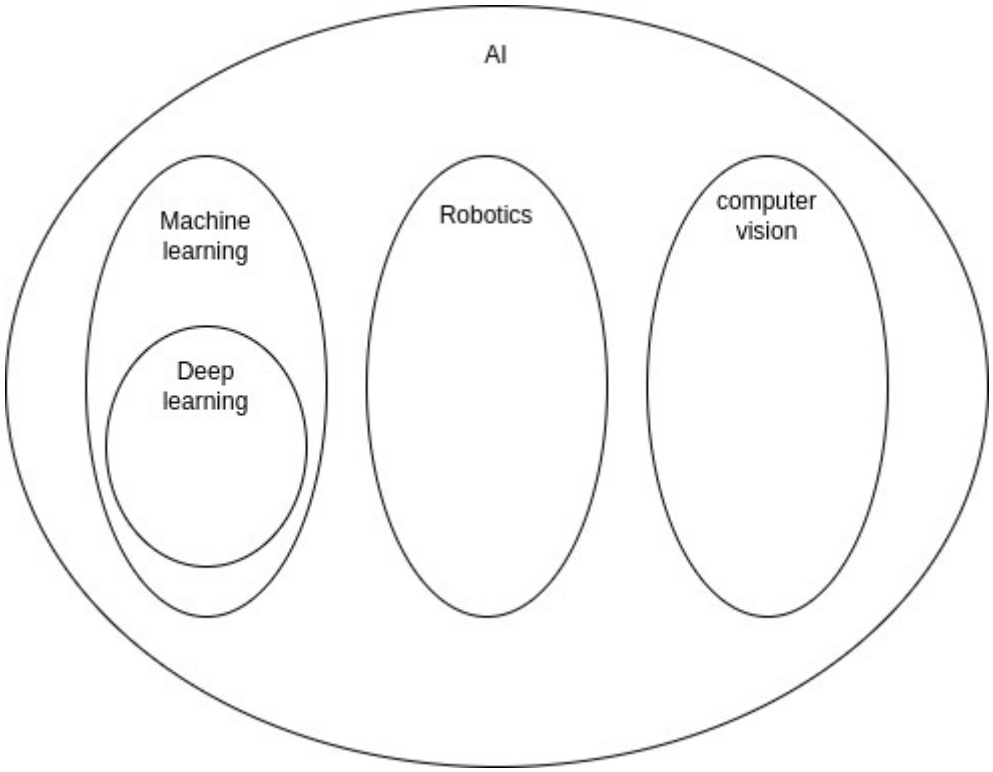


Figura 12: Intelligenza artificiale, aree di applicazione

Il **Machine Learning** (ML) è un sottoinsieme dell'intelligenza artificiale che si concentra sulla creazione di algoritmi e modelli che permettano ai computer di **apprendere** dai dati e di migliorare automaticamente le loro performance nel tempo senza essere esplicitamente programmati per ogni

singolo compito. In altre parole, il machine learning consente ai sistemi di "imparare" dai dati storici, identificando pattern e relazioni, e di fare previsioni o prendere decisioni basate su nuove informazioni.

Il ML si divide in tre principali categorie:

1. **Apprendimento supervisionato**: L'algoritmo è addestrato su un set di dati etichettati, in cui ogni input è associato a una risposta corretta. L'obiettivo è che il modello impari a prevedere o classificare correttamente nuovi dati in base ai pattern appresi.
2. **Apprendimento non supervisionato**: L'algoritmo cerca di identificare strutture o gruppi nei dati senza l'ausilio di etichette. Viene utilizzato, ad esempio, per il clustering o la riduzione dimensionale.
3. **Apprendimento per rinforzo**: L'algoritmo apprende mediante prove ed errori, ricevendo ricompense o penalità per le azioni intraprese, con l'obiettivo di ottimizzare un comportamento o una strategia nel tempo.

Il **Deep Learning** è una branca avanzata del machine learning che si concentra sull'uso di **reti neurali artificiali profonde** (deep neural networks) per risolvere problemi complessi. Queste reti neurali sono composte da numerosi strati (layers) di neuroni artificiali, ciascuno dei quali elabora una rappresentazione sempre più astratta dei dati in ingresso. Il deep learning è particolarmente potente per compiti che coinvolgono grandi quantità di dati non strutturati, come le immagini, il linguaggio naturale e il suono.

Mentre il **machine learning** si concentra su approcci più tradizionali, che includono una varietà di tecniche per l'analisi dei dati e la previsione, il **deep learning** rappresenta una direzione più avanzata, che sfrutta le reti neurali profonde per risolvere problemi particolarmente complessi. Il deep learning ha il vantaggio di poter gestire e apprendere direttamente da enormi volumi di dati non strutturati, come immagini e testi, senza la necessità di un intervento manuale per la selezione delle caratteristiche. Tuttavia, il deep learning richiede risorse computazionali più significative e può risultare meno interpretabile rispetto ad altre tecniche di machine learning.

Le reti neurali artificiali sono generalmente costituite da tre tipi di strati:

1. **Strato di input (Input layer)**
2. **Strati nascosti (Hidden layers)**
3. **Strato di output (Output layer)**

Ogni neurone nella rete riceve un input, che è il risultato delle informazioni che arrivano dai neuroni degli strati precedenti. Questo input viene pesato (ogni connessione ha un peso associato) e trasformato tramite una funzione di attivazione che determina se il neurone deve attivarsi e trasmettere l'informazione al neurone successivo.

1. **Pesi (Weights)**: Ogni connessione tra neuroni ha un peso, che indica l'importanza di quella connessione. Durante il processo di addestramento, i pesi vengono aggiornati per migliorare la performance della rete.
2. **Bias**: Ogni neurone può anche avere un valore di bias, che serve a spostare l'output di un neurone in modo da migliorare la capacità del modello di adattarsi ai dati.
3. **Funzione di attivazione**: La funzione di attivazione determina la risposta del neurone all'input ricevuto. Alcune funzioni comuni sono:
    - **Sigmoid**: Restituisce un valore compreso tra 0 e 1, utile per problemi di classificazione binaria.
    - **ReLU (Rectified Linear Unit)**: Restituisce 0 se l'input è negativo e l'input stesso se è positivo. È molto utilizzata nelle reti neurali profonde.
    - **Tanh**: Funzione iperbolica che restituisce valori tra -1 e 1

Sintentizzando con un immagine una rete minima avrebbe questo schema dove x è l'input che per un immagine può essere il singolo pixel e per una canzone può essere la singola nota del singolo strumento. W sono i pesi da moltiplicare all'input. N è la rete neurale che potrebbe fare semplicemente anche solo una somma delle varie variabili pesate con il bias. A questa somma si potrebbe applicare una qualsiasi funzione di attivazione per avere un risultato di output in base anche al fatto che ci sia 1 solo output o meno. Il risultato della rete neurale è y.

INPUT	**OUTPUT**	**Esempi di reti**
1	1	Classificazione di immagini (rete tradizionale)
1	N	Generazioni di N possibili canzoni
N	1	Classificazioni di sentimenti
N	N	Traduzione di test

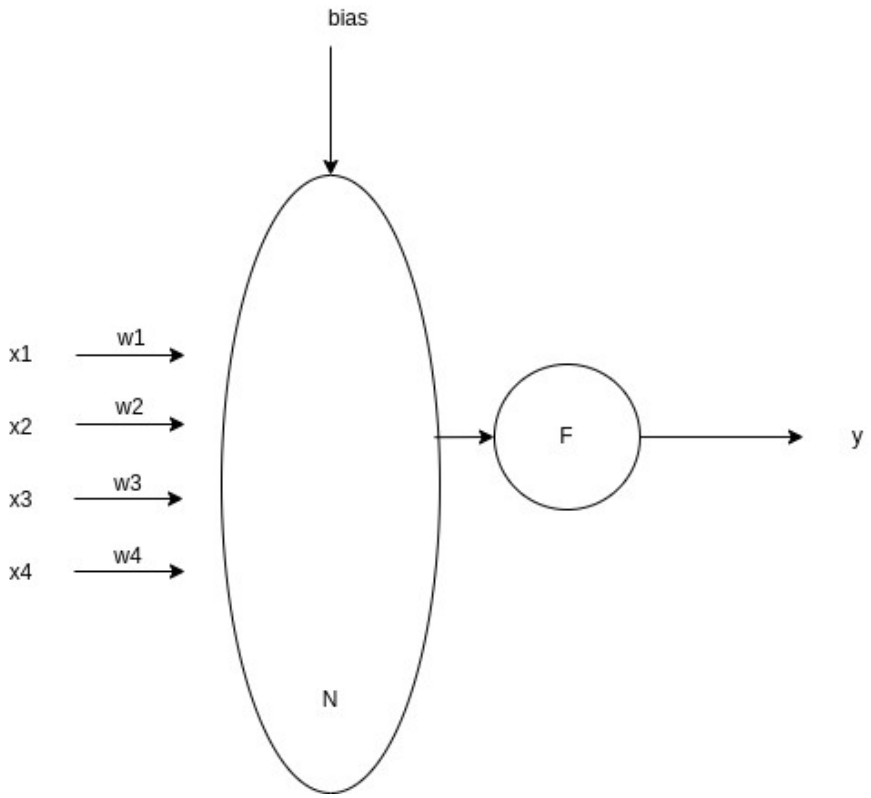


Figura 13: Reti neurali

Il **bias** è un termine aggiuntivo che consente al modello di adattarsi meglio ai dati, migliorando la capacità di apprendere e generalizzare. Matematicamente, il bias può essere visto come un ulteriore parametro che viene aggiunto all'input di una rete neurale. Questo termine permette di spostare la funzione di attivazione, alzando o abbassando la "retta" di decisione, consentendo alla rete di adattarsi meglio a situazioni in cui i dati non passano attraverso l'origine (ovvero, il punto x=0, y=0).

Per comprendere meglio, possiamo considerare una rete neurale come una funzione che mappa gli input x a un output y. In una rete senza bias, questa funzione potrebbe essere semplicemente una retta (nel caso di una rete a singolo strato) della forma:

$$y=ax$$

dove a è il peso (o coefficiente). Tuttavia, l'introduzione di un termine di bias b consente alla funzione di essere spostata lungo l'asse delle ordinate, modificando l'equazione in:

$$y=ax+b$$

In questo modo, il bias b agisce come un **intercetta**, ovvero come un parametro che **sposta la retta** in alto o in basso, permettendo alla rete di adattarsi meglio ai dati anche quando l'origine non è rappresentativa della distribuzione degli input.

Nella pratica delle reti neurali, il bias viene trattato come un **ulteriore peso** associato a un "input costante" pari a 1. La rete impara i pesi associati sia agli input reali sia al bias durante la fase di addestramento. Di fatto, il bias funziona come una costante che, insieme ai pesi, influenza il risultato dell'output della rete, migliorando la capacità di apprendimento della rete stessa.

Addestramento attraverso backpropagation

Il principio fondamentale della retropropagazione (backpropagation) consiste nel calcolare i gradienti della funzione di perdita rispetto ai pesi e ai bias della rete neurale. Questo processo avviene attraverso la differenziazione della funzione di perdita per ciascun parametro del modello. La retropropagazione si svolge in due fasi principali:

1. **Passaggio in avanti (Forward Pass)**: Durante questa fase, gli input vengono passati attraverso i vari layer della rete, dove vengono elaborati per calcolare gli output a ogni nodo. Alla fine del passaggio, si ottiene un risultato che viene confrontato con il valore target, utilizzando una funzione di perdita per misurare l'errore della previsione.
2. **Passaggio all'indietro (Backward Pass)**: Una volta calcolato l'errore (ossia la differenza tra l'output predetto e quello desiderato), il gradiente di questo errore viene propagato all'indietro attraverso i layer. In questa fase si calcolano i gradienti della funzione di perdita rispetto ai pesi e ai bias, utilizzando il teorema della catena. Questi gradienti indicano come ciascun parametro ha contribuito all'errore complessivo. Successivamente, i gradienti vengono usati per aggiornare i pesi e i bias tramite un algoritmo di ottimizzazione, come il gradiente discendente.

Lo scopo di questo processo è minimizzare la funzione di perdita, trovando i pesi e i bias ottimali che permettano al modello di fare previsioni più accurate. Per farlo, si definisce una **funzione di costo** (o loss function), che rappresenta la quantità di errore che il modello commette durante la previsione. Il **gradiente discendente** è un metodo numerico usato per trovare il minimo di questa funzione, cercando di ridurre progressivamente l'errore del modello.

Il comportamento del gradiente discendente dipende da un parametro chiave: la **learning rate** (tasso di apprendimento). Se il valore del tasso di apprendimento è troppo alto, il modello rischia di saltare oltre il minimo della funzione di perdita, senza mai convergere correttamente. Se invece è troppo basso, l'ottimizzazione diventa estremamente lenta, richiedendo molte iterazioni per trovare una soluzione adeguata. La scelta di un learning rate ottimale è cruciale per l'efficienza del training.

Il training di una rete neurale avviene in **epoche**, che rappresentano i cicli completi di addestramento. In ogni epoca, i pesi e i bias vengono aggiornati progressivamente, affinando il

modello per migliorare le previsioni. Il numero di epoche è un **iperparametro** che deve essere scelto attentamente per evitare il **overfitting** (quando il modello si adatta troppo ai dati di addestramento, perdendo la capacità di generalizzare su nuovi dati) o l'**underfitting** (quando il modello è troppo semplice e non riesce a catturare correttamente la relazione nei dati).

Il bilanciamento tra underfitting e overfitting è essenziale per ottenere un modello che abbia una buona capacità di generalizzazione, ossia che sia in grado di fare previsioni accurate anche su dati mai visti prima.

Durante la progettazione e l'addestramento di una rete neurale, è importante tenere conto di diversi fattori pratici:

- **Latenza del training**: Il tempo che la rete impiega per addestrarsi su un dato set di dati.
- **Accuratezza del risultato**: La capacità del modello di fare previsioni corrette.
- **Potenza di calcolo e consumo energetico**: La necessità di risorse computazionali per addestrare il modello, che è spesso direttamente proporzionale all'accuratezza e alla complessità del modello stesso.

Questi fattori sono interconnessi: ad esempio, modelli più complessi richiedono più potenza di calcolo e maggiore tempo di addestramento, ma tendono anche a ottenere prestazioni migliori. L'obiettivo è trovare un equilibrio ottimale che consenta di ottenere una rete ben addestrata, ma al tempo stesso efficiente in termini di risorse.

Tensorflow vs Pytorch

In fase di sviluppo delle reti neurali, si è scelto di utilizzare **TensorFlow** anziché **PyTorch** per diverse ragioni legate alle specifiche esigenze del progetto e alla maturità delle due librerie nel contesto applicativo. TensorFlow è ampiamente utilizzato nell'industria e offre una solida infrastruttura per la **scalabilità** e l'**ottimizzazione** delle prestazioni, risultando particolarmente adatto per applicazioni in produzione che richiedono un elevato grado di efficienza e stabilità. Inoltre, la sua integrazione con strumenti di supporto come **TensorFlow Serving** per la gestione dei modelli in produzione e **TensorFlow Lite** per dispositivi mobili lo rende una scelta privilegiata per implementazioni su larga scala.

D'altro canto, **PyTorch**, pur essendo una libreria estremamente potente e flessibile, è più orientata alla ricerca scientifica, dove la **dinamicità** del suo grafico computazionale e la facilità nell'eseguire debug e sperimentazioni rapide lo rendono particolarmente utile. Tuttavia, per un ambiente industriale che richiede robustezza, performance ottimizzate e supporto per il deployment su larga scala, TensorFlow rappresenta una soluzione più consolidata e adatta alle necessità del progetto.

Queste considerazioni hanno portato alla scelta di **TensorFlow** come framework di sviluppo principale, dato il suo utilizzo diffuso nel settore e la sua capacità di supportare efficacemente il ciclo completo di sviluppo, training e deployment delle reti neurali.

Tipologie di Apprendimento delle Reti Neurali

Il processo di addestramento di una rete neurale è finalizzato a determinare i **pesi** e i **bias** che minimizzano l'errore del modello rispetto ai dati di addestramento. In base al tipo di feedback disponibile durante il processo di addestramento, esistono diverse modalità di apprendimento delle reti neurali. Le principali tipologie di **allenamento** sono:

1. Apprendimento Supervisionato (Supervised Learning)

Nell'**apprendimento supervisionato**, la rete neurale viene addestrata su un dataset che contiene sia gli **input** che gli **output** desiderati. Ogni esempio nel dataset è costituito da una coppia (x_i,y_i), dove x_i è un vettore di input e y_i è il corrispondente output target. Il compito dell'algoritmo di apprendimento è quello di **minimizzare l'errore** tra le previsioni del modello e gli output desiderati.

2. Apprendimento Non Supervisionato (Unsupervised Learning)

L'**apprendimento non supervisionato** è una modalità in cui la rete neurale non riceve etichette di output. In altre parole, l'algoritmo deve scoprire **pattern nascosti** o **strutture** nei dati senza un obiettivo esplicito predefinito. Gli algoritmi di apprendimento non supervisionato cercano di organizzare i dati in modo significativo, per esempio, raggruppandoli in **cluster** o riducendo la dimensionalità.

Alcuni approcci tipici includono:

- **Clustering** (ad esempio, il **K-means clustering**),
- **Riduzione della dimensionalità** (ad esempio, **Principal Component Analysis** o **autoencoder**).

L'apprendimento non supervisionato è utile quando non è disponibile un dataset etichettato o quando si vuole esplorare e scoprire strutture intrinseche nei dati.

3. Apprendimento per Rinforzo (Reinforcement Learning)

L'**apprendimento per rinforzo** (RL) è un paradigma in cui un **agente** interagisce con un ambiente e **impara** a prendere decisioni ottimali per massimizzare una **ricompensa cumulativa** nel lungo termine. In RL, l'agente non ha accesso a etichette fisse come nel caso dell'apprendimento supervisionato, ma riceve un segnale di ricompensa o punizione dopo ogni azione compiuta.

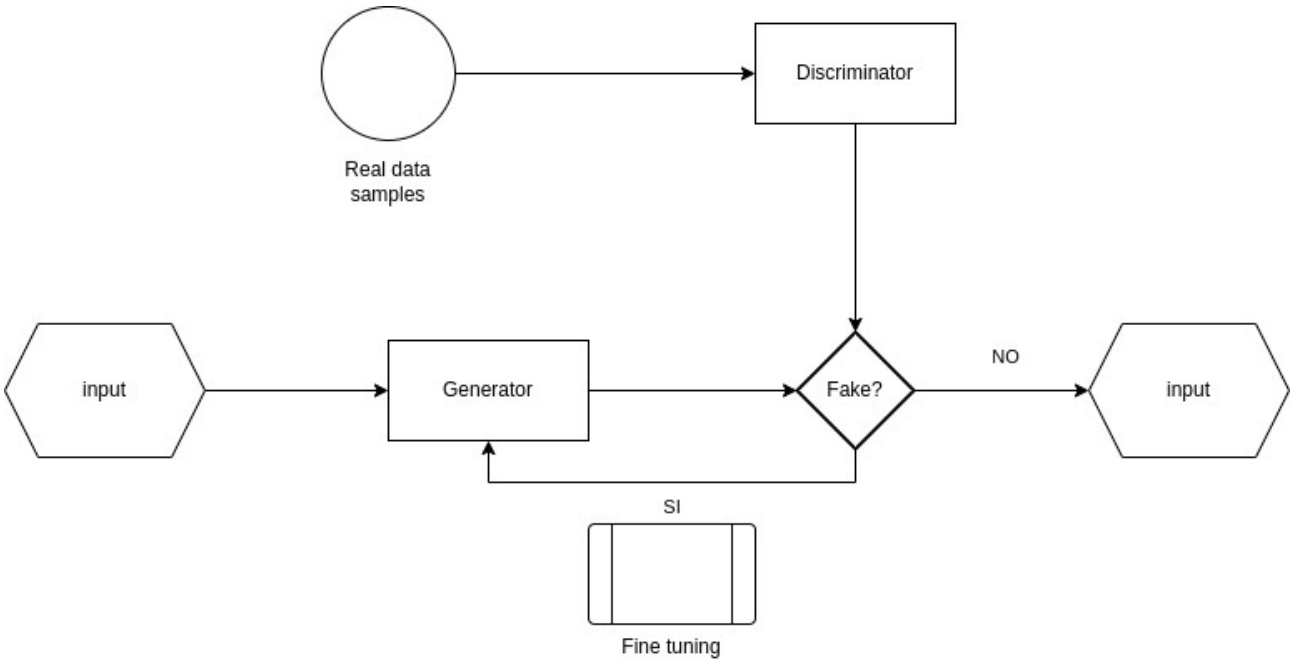


Figura 14: Gan, schema concettuale

La rete è composta da 2 componenti: il discriminator ed il generator. Il discriminator è addestrato con esempi veri e serve a decidere se il generatore sta generando dati compatibili. Quindi definito il problema sul discriminator e creata la rete del generator si eseguono i seguenti passi di massima:

1. training del modello del discriminator sui dati reali
2. genero fake input sul generator
3. insegno al discriminatore a riconoscere i fake input
4. insegno al generatore a generare input accettati dal discriminatore

Tra i principali vantaggi di tale approccio si annoverano:

- la capacità di generare risultati di elevata qualità;
- l'assenza della necessità di un'associazione esplicita tra input e output, il che ne consente la classificazione come un metodo di apprendimento non supervisionato (unsupervised learning);
- la versatilità dell'output prodotto.

Tra gli svantaggi principali associati alle Generative Adversarial Networks (GAN) si rilevano:

- **Instabilità durante il processo di training**: la natura competitiva tra il generatore e il discriminatore può portare a difficoltà nel raggiungere una convergenza stabile, con il rischio di oscillazioni nel comportamento del modello;
- **Alto costo computazionale**: il processo di addestramento delle GAN richiede significative risorse computazionali, in particolare per modelli di grande complessità e per set di dati di ampie dimensioni;
- **Probabilità di overfitting**: in presenza di dati limitati o di modelli non sufficientemente regolarizzati, le GAN sono suscettibili al fenomeno dell'overfitting, con una generazione di output che potrebbe non generalizzare adeguatamente a nuovi dati;
- **Difficoltà nella giustificazione dei risultati**: data la natura non interpretabile delle architetture GAN, risulta complesso fornire spiegazioni chiare e trasparenti riguardo le decisioni prese dal modello, limitando la sua applicabilità in contesti che richiedono spiegabilità.

Per la generazione di contenuti tra i quali la musica ci focalizzeremo su reti per il deep learning in cui l'apprendimento sarà non supervisionato oppure per rinforzo.

Classi di modelli ad apprendimento non supervisionato

Il primo esempio che faccio sono le reti CNN in quanto sono le prime reti nate con apprendimento non supervisionato.

Le **Convolutional Neural Networks (CNN)** sono una classe di reti neurali artificiali particolarmente efficaci per compiti di riconoscimento e classificazione di immagini, ma anche per altri tipi di dati che presentano una struttura spaziale o temporale (ad esempio, segnali audio o video). Le CNN si ispirano al funzionamento del sistema visivo umano e sono composte da più strati (layer) che apprendono automaticamente caratteristiche gerarchiche dei dati.

Le CNN sono costituite da diversi strati che si specializzano nell'estrazione di caratteristiche a vari livelli di astrazione. I principali tipi di layer (strati) sono:

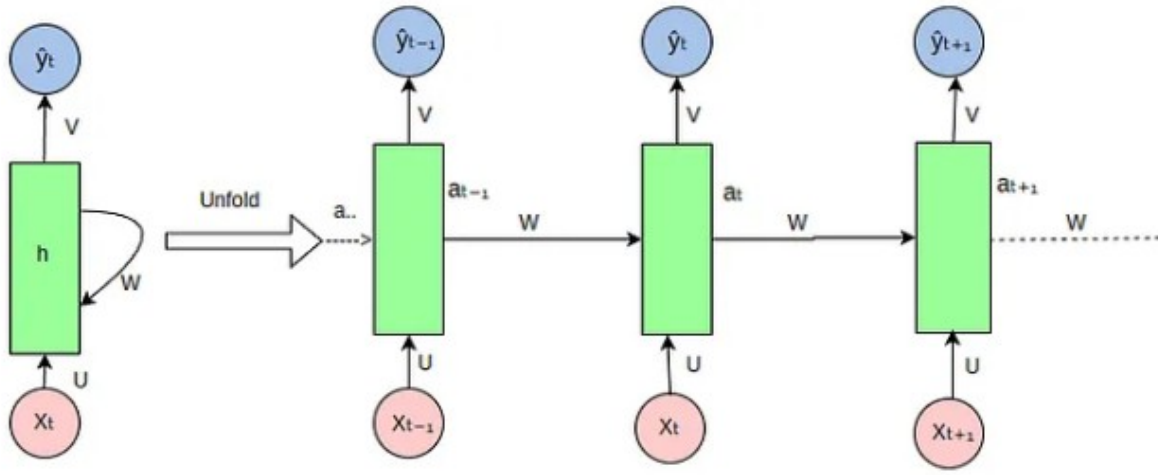
- **Strato di Convoluzione (Convolutional Layer):** Il cuore della CNN è lo strato di convoluzione, che utilizza filtri (o kernel) per scansionare (convolvere) l'immagine di input e produrre una mappa di attivazione (feature map). Questi filtri sono apprendibili durante il processo di addestramento e consentono alla rete di estrarre caratteristiche come bordi, angoli, forme, texture e, in strati successivi, caratteristiche sempre più complesse.
- **Strato di Pooling (Pooling Layer):** Lo strato di pooling ha la funzione di ridurre la dimensione spaziale (larghezza e altezza) della mappa di attivazione, riducendo così il numero di parametri e la complessità computazionale, oltre a rendere la rete meno sensibile

alle variazioni di posizione. I due tipi più comuni di pooling sono il *max pooling* (prende il valore massimo da una regione) e il *average pooling* (calcola la media dei valori).

Quindi, mentre lo strato di convoluzione **espande** la complessità dei dati, lo strato di pooling **compatta** e semplifica tali dati, consentendo alla CNN di essere più efficiente nell'analizzare informazioni rilevanti e ridurre il carico computazionale.

In sintesi, **convoluzione = estrazione di caratteristiche** e **pooling = riduzione della complessità**.

Le **Recurrent Neural Network (RNN)** invece sono classi di modelli progettate per gestire dati sequenziali o temporali. Mantengono le informazioni da passaggi precedenti della sequenza grazie alla loro memoria interna.



Al passo a_{t-1} produrrà un output diverso rispetto al passo a_t

Una sottoclasse di RNN è la classe **Long Short Term Memory (LSTM)** in cui lo step successivo è determinato dal passo in cui si è e dall'input X_t il quale sceglie la parte importante del passo precedente e trascura la parte meno significativa.

L'ultima classe di modelli che voglio trattare invece è l'autoencoder. Un **autoencoder** è un tipo di rete neurale progettata per apprendere una rappresentazione compatta (o codifica) dei dati in input, al fine di poterli successivamente ricostruire in maniera fedele. L'obiettivo di una rete di questo tipo è acquisire una mappatura di alta qualità in uno spazio a dimensione ridotta, dalla quale è possibile generare dati variegati, mantenendo le caratteristiche principali degli input originali. In sostanza,

l'autoencoder è in grado di **apprendere una rappresentazione comprimibile** che permette di **generare nuove istanze di dati** simili a quelle osservate durante l'addestramento.

Dall'immagine sotto si nota una prima parte che codifica le informazioni e una seconda parte che le decodifica. La decodifica non è la copia dell'input, ma l'output della rete generativa.

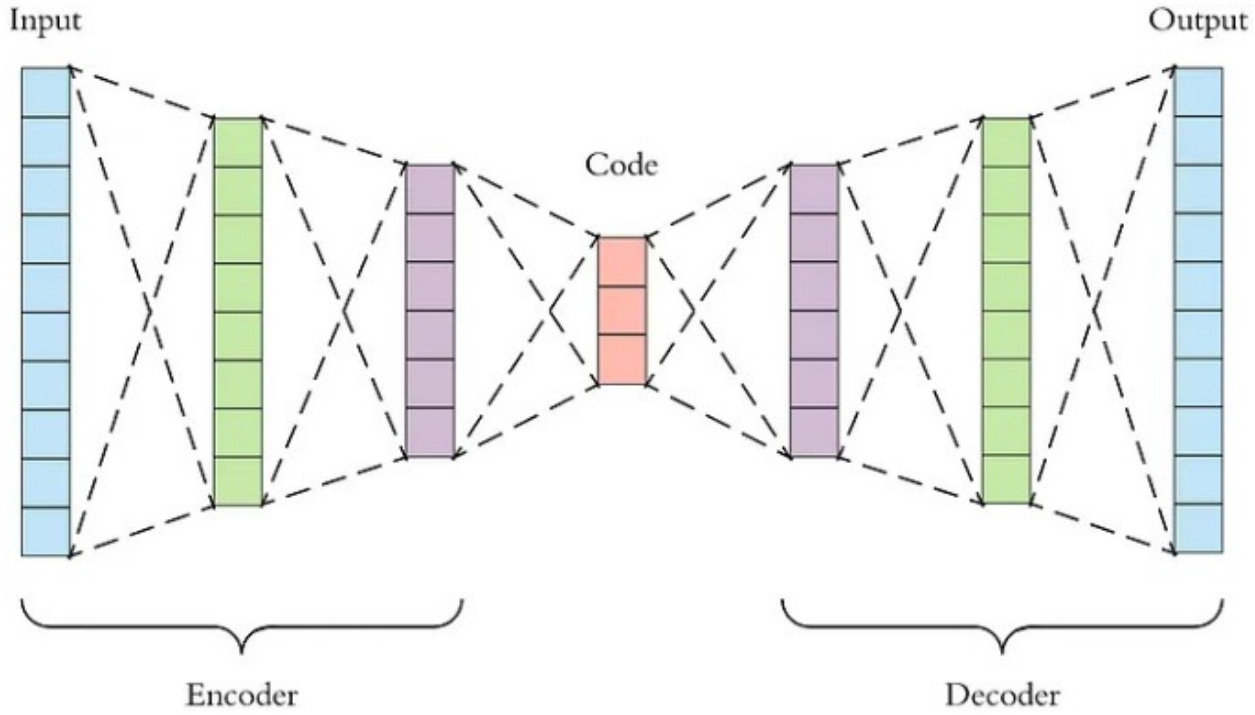


Figura 16: Vae, encoding e decoding

Il **Variational Autoencoder (VAE)** è una particolare variante degli autoencoder che integra tecniche di inferenza bayesiana per generare nuovi dati. A differenza degli autoencoder tradizionali, il VAE assume che la codifica dei dati avvenga in uno spazio latente probabilistico, dove le variabili latenti non sono deterministiche, ma seguono una distribuzione probabilistica (tipicamente una distribuzione normale multivariata). Durante il processo di addestramento, il VAE cerca di ottimizzare una funzione obiettivo che bilancia due termini: la ricostruzione accurata dei dati e la regolarizzazione della distribuzione delle variabili latenti, affinché questa sia vicino a una distribuzione predefinita (ad esempio, una normale multivariata standard). Questo approccio consente al VAE non solo di comprimere e ricostruire i dati, ma anche di generare nuovi campioni variegati e realistici, aprendone l'uso per compiti di generazione di dati e modellazione probabilistica.

Bibliografia

1. tfa insegnante di sostegno nella scuola dell’infanzia e primaria, Edizioni Simone
2. Design specific user interfaces for people with down syndrome using suitable WCAG 2.0 guidelines, Lucia Alonso - Virgos1 · Jordán Pascual Espada1 · Luis Rodríguez Baena1 · Rubén González Crespo
3. https://www.worldscientific.com/doi/abs/10.4015/S1016237218500072
4. https://www.researchgate.net/profile/Laberiano-Andrade-Arenas/publication/349745106_Impact_of_Mobile_Applications_for_a_Lima_University_in_Pandemic/links/614b731d3c6cb31069874b3a/Impact-of-Mobile-Applications-for-a-Lima-University-in-Pandemic.pdf
5. https://dl.acm.org/doi/abs/10.1145/2501988.2502055
6. https://www.seashelltrust.org.uk/therapy-and-nursing/
7. https://dsq-sds.org/index.php/dsq/article/view/5968/4703
8. La sindrome di down di Stefano Vicari, 9788815230140
9. https://voices.no/index.php/voices/article/view/3405/3541
10. http://www.smj.org.sg/sites/default/files/4312/4312le1.pdf
11. https://www.youtube.com/watch?v=RKK7wGAYP6k&t=20s
12. https://www.missouristate.edu/LOGOS/_Files/logos_vol4_full.pdf#page=45
13. https://www.jneurosci.org/content/jneuro/30/45/14943.full.pdf
14. https://academic.oup.com/ptj/article/87/10/1399/2742283
15. Children with down syndrome and music: A parental description of their experience in music. Daudt, Alyssa, https://researchdiscovery.drexel.edu/view/pdfCoverPage?instCode=01DRXU_INST&filePid=13321508890004721&download=true
16. https://www.youtube.com/watch?v=x7pWgrXNsbo&t=172s svs
17. https://www.icf-casestudies.org/introduction/introduction-to-the-icf/the-integrative-bio-psycho-social-model-of-functioning-disability-and-health