Università degli Studi di Padova
Dipartimento di Scienze Statistiche
Corso di Laurea Triennale in

Statistica per le Tecnologie e le Scienze

METODI STATISTICI PER IL RICONOSCIMENTO AUTOMATICO DI STRUMENTI MUSICALI

Relatore Prof. Carlo Ferrari
Dipartimento di Ingegneria dell'Informazione

Laureando: Alberto Cera
Matricola n. 2004553

Anno Accademico 2023/2024

Abstract

Il seguente lavoro pone come obbiettivo lo sviluppo di un sistema di riconoscimento e classificazione degli strumenti musicali mediante l'impiego di tecniche di apprendimento automatico. Nel corso dello studio, ho confrontato le performance di due modelli di machine learning, Support Vector Machine (SVM) e k-Nearest Neighbors (KNN), al fine di valutare la loro efficacia nella classificazione degli strumenti musicali. I risultati forniti contribuiscono a una comprensione approfondita delle prestazioni di tali approcci nel contesto specifico del riconoscimento degli strumenti musicali basato su dati audio.

Alla mia famiglia

Indice

Capitolo 1

Presentazione del problema

Questo progetto consiste nell'implementazione di un sistema di riconoscimento di strumenti musicali per tracce mono-strumentali, e nel confronto dei risultati ottenuti con diverse strutture. Una particolare attenzione nel seguente studio è stata posta nelle fasi di pre-processing, estrazione dei dati di addestramento e di valutazione del sistema.
Il seguente progetto è stato svolto in Python.

1.1 Contesto e Settore di Ricerca

Negli ultimi anni c'è stato un notevole interesse e sviluppo nel campo della ricerca sui dati musicali, grazie ai progressi delle tecniche di data mining e di estrazione dei dati. Questo ha alimentato una crescente attività di studio focalizzata su vari aspetti, tra cui il reperimento di musica basata sui contenuti, la classificazione dei generi musicali e l'analisi delle performance strumentali, ma anche la produzione automatizzata di musica tramite intelligenze artificiali.

Particolarmente rilevante è l'attenzione dedicata al riconoscimento e alla classificazione dei vari strumenti musicali. Questo settore di ricerca ha suscitato grande interesse per le sue molteplici applicazioni e implicazioni, che vanno ben oltre l'identificazione degli strumenti in una registrazione. Ad esempio, le tecniche di rilevamento degli strumenti sono state impiegate per analizzare passaggi solistici,

contribuendo così a una migliore la comprensione dei differenti stili musicali, fornendo un supporto prezioso per le lezioni di musicologia. Inoltre, queste tecnologie sono state integrate con successo in software di editing audio, consentendo agli utenti di manipolare in modo più efficace le tracce strumentali all'interno di una registrazione. Questo ha aperto nuove opportunità nel campo della produzione musicale e della post-produzione audiovisiva, migliorando la qualità e la precisione delle produzioni finali.

Oltre all'editing audio, il rilevamento degli strumenti ha trovato applicazione nel recupero e nella trascrizione di registrazioni audio e video. Identificare automaticamente gli strumenti presenti in una registrazione può semplificare notevolmente il processo di trascrizione musicale, consentendo ai musicisti di ottenere partiture più accurate e di preservare il patrimonio musicale in modo più efficiente.

Infine, nel contesto dell'industria musicale, le tecnologie di rilevamento degli strumenti stanno rivoluzionando la progettazione e l'implementazione di strumenti musicali digitali e software audio [1].

Attualmente, uno dei problemi principali che si pone in questo campo è l’identificazione di strumenti considerati complessi come strumenti elettronici e sintetizzatori, le cui frequenze si confondono spesso con strumenti acustici.

Il seguente progetto pone la sperimentazione di un sistema di riconoscimento di strumenti acustici. Il suo utilizzo potrà essere comunque adeguato a strumenti più complessi da riconoscere.

1.2 Alcune Proposte di Risoluzione del Problema

Negli ultimi anni sono state proposte diverse metodologie di risoluzione. L'attenzione in particolare viene posta in tre passaggi fondamentali: estrazione dei dati, riduzione della dimensionalità del dataset e scelta dell'algoritmo di classificazione supervisionata degli strumenti musicali.

Un approccio proposto in uno studio condotto presso "Istitute of Information Technology Lodz, Poland" dai Professori Dominika Szeliga e Pawel Tarasiuk, propone un iniziale trasformazione dei suoni appartenenti al dataset di addestramento nelle relative immagini degli spettrogrammi in scala logaritmica. Successivamente le immagini ottenute (in scala di grigi) sono state utilizzate per addestrare un modello basato su rete neurale Convoluzionale. Il dataset utilizzato è il dataset IRMAS, la cui composizione sarà approfondita successivamente. Questo studio in particolare pone attenzione al processo di una rete neurale Convoluzionale e confronta i risultati, in termini di accuratezza e precisione, ottenuti variando il numero di *layer* del modello e il numero di unità presenti nell'insieme di addestramento [2].

Un approccio alternativo è stato proposto da uno studio condotto presso il "Center for Engineering, Modeling and Applied Social Sciences (CECS) Federal University of ABC (UFABC) Santo Andre, Brazil", dai Professori Alexandre Lucena, Caroline Moraes, Kenji Nose-Filho e Denis Fantino. In questo studio, condotto sempre sul dataset IRMAS, da ogni traccia sono stati estratti i relativi spettrogrammi Mel (scala di percezione dell'altezza del suono).

L'obiettivo di questo studio risulta essere il confronto, a parità di condizioni di partenza, tra due algoritmi di classificazione come il *Support Vector Machine* e il *Multilayer Perceptron*. I risultati, espressi in termini di accuratezza e F1-Score, indicano prestazioni più soddisfacenti per il modello basato su *Multilayer Perceptron* [3].

Un ulteriore studio, condotto presso il "Department of E&TC JSPM's RSCOE, Tathawade,India", dalla Professoressa Shilpa Sonawane, introduce l'estrazione dei dati dalle tracce audio presenti nel dataset IRMAS non più mediante grafici delle frequenze, bensì mediante alcuni indici relativi alla traccia, come il *Log-Attack Time*, il *Temporal centroid*, alcuni indici relativi invece allo spettrogramma come lo *Spectral Centroid*. Come classificatore viene utilizzato un modello basato sempre su Support Vector Machine [4].

Un'estensione di questi studi si è raggiunta con uno studio svolto presso "National and Kapodistrian University of Athens, Department of Informatics and Telecommunications" dai Professori Giorgos Mazarakis, Panagiotis Tzevelekos, Georgios Kouroupetroglou. Questo lavoro si basa essenzialmente su un iniziale elaborazione del segnale codificato nel tempo (per estrarre le feature dai file audio) e l'applicazione di modelli di classificazione basati su *Fast Artificial Neural Network*. Lo studio viene condotto in un dataset più complesso del dataset IRMAS, cioè un dataset di 490 file audio, ciascuno contenente una singola nota, riprodotta da un sintetizzatore, in grado di simulare i 19 strumenti musicali considerati nello studio [5].

Lo studio "Musical instrument recognition on solo performance" di Slim Essid, Gaël Richard, Bertrand David, propone una riduzione

della dimensionalità del dataset iniziale, prima dell'addestramento del modello di classificazione, mediante *PCA*. Dai file audio, contenenti le registrazioni dei singoli strumenti, vengono estratte alcuni coefficienti che riassumono l'andamento dello spettrogramma. Questi vengono inseriti all'interno di un dataset, il quale viene ridotto di dimensionalità tramite PCA e viene utilizzato per addestrare due modelli, uno basato su Support Vector Machine, l'altro basato su un modello a *Miscela Gaussiana*. L'utilizzo della PCA sembra portare ad un miglioramento delle prestazioni per entrambi i modelli testati [6].

Un altro approccio viene proposto da uno studio condotto presso il "Department of Computer Science and Engineering, Annamalai University, Chidambaram, Tamilnadu, India", da S. Prabavathy, V. Rathikarani, P. Dhanalakshmi. In questo studio le tracce audio del dataset IRMAS vengono trasformate in spettrogrammi MEL (in bianco e nero). Queste immagini, in seguito, vengono scomposte in pixel e inserite all'interno di un dataset. La riduzione del dataset viene applicata mediante una rete neurale Convoluzionale a 22 livelli, ovvero la *GoogleNet*. Come modello di classificazione vengono considerati un modello basato su *SVM* e un modello basato su *K-Nearest Neighbors* [7].

Nello studio che presenterò successivamente, l'estrazione dei dati avviene mediante elaborazione dei pixel delle immagini degli spettrogrammi Mel di ogni traccia; successivamente ho voluto valutare l'effetto della riduzione di dimensionalità mediante PCA considerando l'utilizzo di due classificatori, uno basato su SVM e uno basato su KNN.

Capitolo 2

Discussione Teorica

In questo capitolo mi soffermerò sull'introdurre dal punto di vista teorico e concettuale alcune tecniche statistiche, utilizzate nel codice. Mi soffermo sulla *Principal Component Analysis* (PCA), utilizzata in fase di pre-processing per ridurre la dimensionalità dei dati e permettere una maggiore efficienza dell'addestramento dei modelli di classificazione utilizzati; in seguito verranno introdotte le due tipologie di modelli di previsione confrontati, ovvero il *Support Vector Machine* e il *K-Nearest Neighbors*.

2.1 Pre-Processing: Principal Component Analysis

La Principal Component Analysis (PCA) è uno strumento statistico impiegato per ridurre la complessità dei dati, preservando le informazioni essenziali. Teoricamente, la PCA si avvale del calcolo della matrice di covarianza dei dati originali, che cattura le relazioni statistiche tra le diverse variabili. Successivamente, vengono identificati gli autovettori e gli autovalori di questa matrice. Gli autovettori indicano le direzioni lungo le quali i dati variano di più, mentre gli autovalori quantificano l'importanza di queste direzioni in termini di varianza spiegata dei dati. Le componenti principali vengono ottenute ordinando gli autovettori in base al valore degli autovalori, in ordine decrescente. Ogni componente principale rappresenta una combinazione lineare delle variabili originali e

rappresenta un asse nel nuovo spazio delle variabili. L'obiettivo della PCA è massimizzare la varianza dei dati lungo queste nuove direzioni, semplificando così la struttura dei dati e consentendo una migliore comprensione e interpretazione [8]. In sintesi, la PCA consente di proiettare i dati su un sottospazio di dimensioni ridotte, mantenendo al contempo la variazione più significativa nei dati originali (Fig. 2.1).

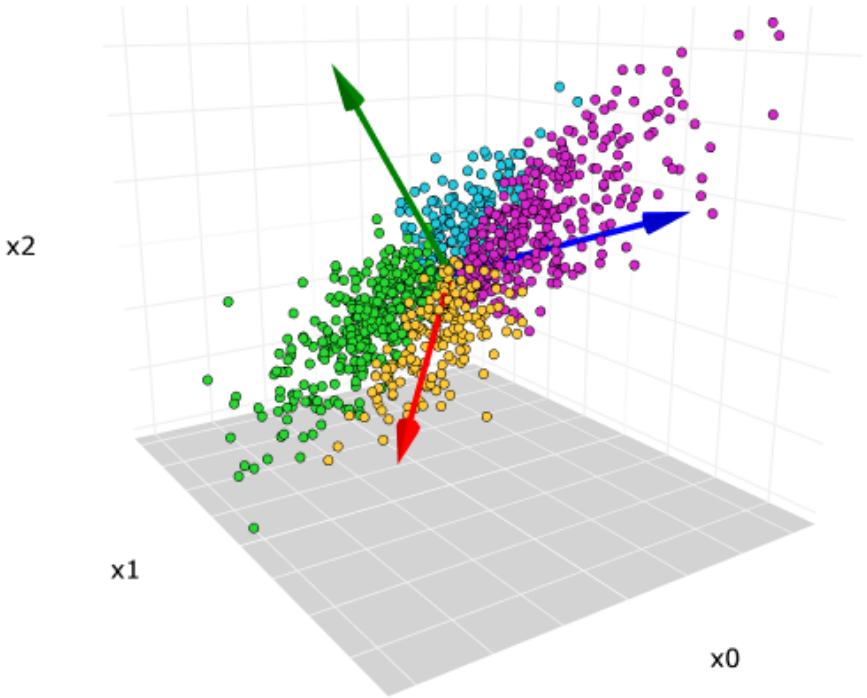


Fig. 2.1: Gli assi blu, verde e rosso rappresentano
le nuove direzioni principali individuate dalla PCA.

2.2 Classificazione mediante Support Vector Machine

Il Support Vector Machine (SVM) è un algoritmo di apprendimento supervisionato ampiamente impiegato per la classificazione e la regressione. La sua efficacia deriva dalla sua capacità di gestire sia problemi lineari che non lineari attraverso la creazione di un iperpiano di separazione ottimale tra le diverse classi dei dati.

L'obiettivo principale della SVM è massimizzare il *margine* tra le istanze di diverse classi, dove il margine rappresenta la distanza tra l'iperpiano di decisione e i punti più vicini di ciascuna classe, noti come vettori di supporto. Questo iperpiano è definito in modo da massimizzare la distanza tra i vettori di supporto, garantendo così la robustezza e la generalizzazione del modello.

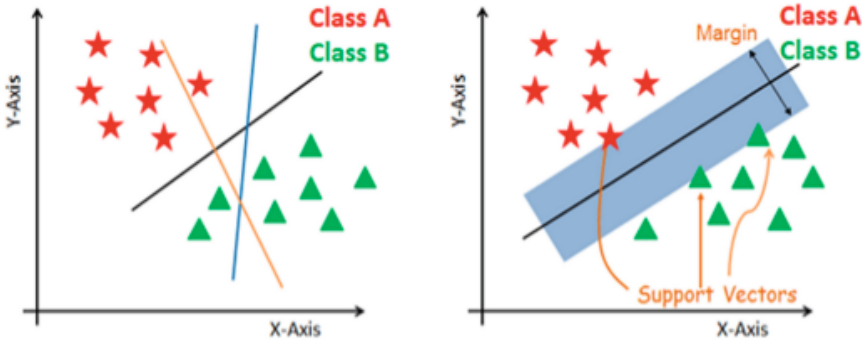


Fig. 2.2: Processo di individuazione dell'iperpiano di una SVM

Per affrontare problemi non lineari, la SVM utilizza il "kernel trick", una tecnica che trasforma lo spazio dei dati in uno spazio di dimensioni superiori, consentendo la separazione lineare delle classi in uno spazio più ampio. Questo approccio permette alle SVM di gestire con successo dati complessi e non lineari.

In sintesi, la SVM si presenta come un algoritmo flessibile e potente, utilizzato in svariate applicazioni come la classificazione di testi e immagini, il riconoscimento di pattern e la bioinformatica. La sua versatilità e capacità di affrontare sia problemi lineari che non lineari la rendono una scelta popolare in diversi contesti di apprendimento automatico [8].

2.2.1 Classificazione Multi-classe mediante Support Vector Machine

La SVM rappresenta una tecnica di classificazione binaria.
Di conseguenza si è presentata la necessità, per adempiere al nostro problema di classificazione, di estendere il suo utilizzo alla classificazione multi-classe.
Questa estensione si applica secondo l'approccio "One-vs-One".
L'idea alla base è quella di creare un classificatore binario per ogni coppia di classi possibili, quindi per K classi verranno creati K*(K-1)/2 classificatori distinti.
Quando si tratta di classificare una nuova istanza, ciascun classificatore emette una predizione. La classe che viene predetta più frequentemente tra tutti i classificatori diventa la classe finale assegnata all'istanza.
Un'alternativa è rappresentata dall'approccio "One-vs-All". Per affrontare un problema di classificazione di K classi, vengono addestrati K classificatori binari separati, ognuno dei quali distingue una delle classi target dalle restanti.

In questo lavoro si è preferita la strategia “One-vs-One” per i seguenti motivi:

- Le prestazioni di algoritmi come il Support Vector Machine degradano rapidamente con l’aumentare dei dati, specie per i tempi di training richiesti. Pertanto, è preferibile allenare molteplici classificatori su sezioni contenute di dati.

- Il problema in questione riguarda un numero di classi contenuto; quindi, non risulta un problema l’utilizzo di un numero di classificatori maggiore.

- Non soffre di problemi di sbilanciamento delle classi (questo problema riguarda soprattutto l’approccio “OvA”, in quanto vi sono K classificatori, ciascuno che distingue una classe dalle rimanenti, e in caso vi siano classi meno numerose, il riconoscimento di queste, nel confronto con tutto l’insieme delle altre classi, solitamente risulta più complicato).

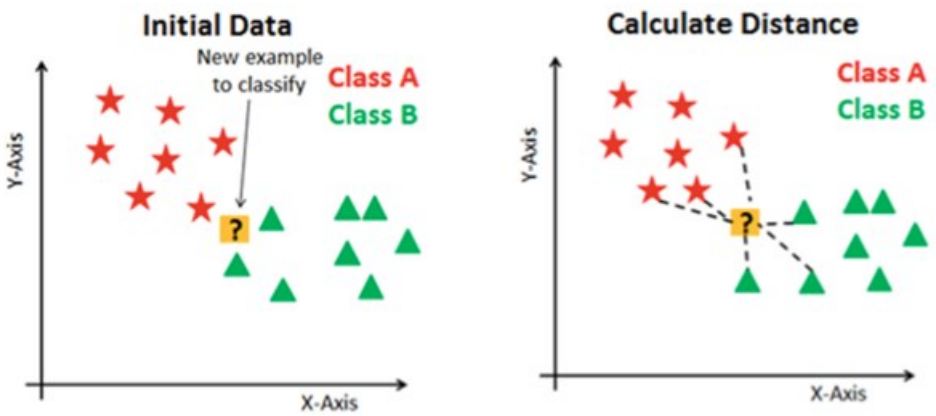
- Permette la formazione di classificatori specifici per la distinzione di ciascuna classe; questo potrebbe portare prestazioni migliori rispetto a classificatori generici [9].

2.3 Classificazione mediante: K-Nearest Neighbors

L'algoritmo k-Nearest Neighbors (k-NN) costituisce un metodo di classificazione o regressione basato sulla misura di vicinanza tra i dati. Inizialmente, si dispone di un set di dati di addestramento contenente esempi etichettati, e si seleziona il parametro k, che rappresenta il numero di vicini più prossimi da considerare durante il processo di classificazione o regressione.

Successivamente, per una nuova unità da classificare, vengono calcolate le distanze rispetto a tutti i punti del set di addestramento, mediante l'utilizzo della distanza euclidea. I k vicini più prossimi vengono identificati sulla base di tali distanze. Nella fase finale, la classificazione avviene assegnando all'oggetto in questione l'etichetta di classe più frequente tra i k vicini (nel caso di classificazione) o calcolando la media o la mediana dei valori di output dei k vicini (nel caso di regressione).

L'idea chiave sottesa a k-NN è che istanze simili tendono a raggrupparsi nello spazio delle caratteristiche. Pertanto, se un nuovo punto è simile a k punti noti, ci si aspetta che appartenga alla stessa classe o abbia un valore di output simile.



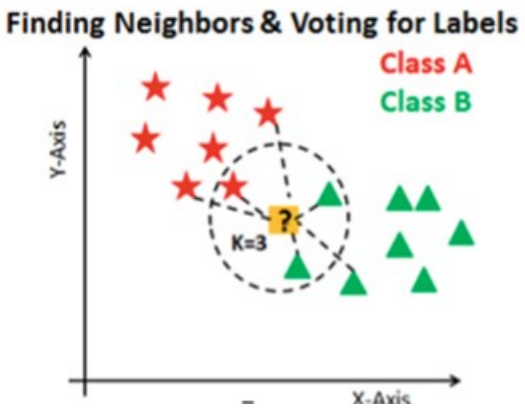


Fig. 2.3: Esempio di addestramento di un modello K-Nearest Neighbors

Alcuni fattori cruciali includono la scelta di k, la metrica di distanza e la gestione della sensibilità a dati con scale diverse. Nonostante l'onerosità computazionale del programma, l'algoritmo k-NN rappresenta un approccio valido per problemi di classificazione o regressione, soprattutto quando la struttura dei dati è complessa e non lineare [8].

Capitolo 3

Il Sistema di Riconoscimento

3.1 Dataset IRMAS

Il dataset utilizzato per l'addestramento e la valutazione dei modelli è il dataset IRMAS (Instrument Recognition in Musical Audio Signal). Questo è un dataset ampiamente utilizzato per l'addestramento e la valutazione di algoritmi di riconoscimento degli strumenti musicali. È stato creato da un team di ricercatori dell'Institute of Systems Engineering and Computer Vision dell'Università di Magdeburgo, in Germania e pubblicato nel 2013.

Il dataset di addestramento usato contiene 6705 registrazioni audio di strumenti musicali di durata compresa tra 5 e 20 secondi. Ogni registrazione audio è associata a una delle 11 categorie di strumenti. Ogni categoria ha un numero simile di esempi nel dataset per garantire un bilanciamento delle classi. Le registrazioni sono fornite nel formato WAV a 16 bit e 44,1 kHz di frequenza di campionamento.
L'annotazione dello strumento predominante di ciascuna traccia è presente sia nel nome della cartella che lo contiene, sia nel nome del file: violoncello (cel), clarinetto (cla), flauto (flu), chitarra acustica (gac), chitarra elettrica (gel), organo (org), pianoforte (pia), sassofono (sax), tromba (tru), violino (vio) e voce cantata umana (voi).

Queste tracce comprendono musica del secolo attuale e di vari decenni del secolo scorso, che differiscono quindi in larga misura per qualità audio. Coprono inoltre una grande variabilità nei tipi di strumenti musicali, negli esecutori, nelle articolazioni e negli stili generali di

registrazione e produzione. Si è cercato inoltre di variare nella composizione del dataset, il più possibile, i generi musicali all'interno della raccolta per evitare di estrarre informazioni relative alle caratteristiche del genere.

Il dataset IRMAS si compone anche di un test set di composizione simile al train set contenente 2874 tracce. Questa parte del dataset contiene estratti di diversi generi musicali occidentali, contenenti diverse strumentazioni. In accordo con gli altri studi citati nel capitolo precedente, ho preferito utilizzare solo il train set del dataset originale, che in seguito ho diviso in train e test set. Questa scelta ha permesso di mantenere una certa similarità tra dataset di addestramento e di test, evitando di inserire brani polifonici, il cui riconoscimento non è l'obiettivo di questo studio [10].

3.2 Estrazione delle Feature dal Dataset

Il primo passaggio dello studio è la trasformazione di tutti i file audio in immagini in scala di grigi di spettrogramma di Mel con risoluzione 480x400. Lo spettrogramma è stato ottenuto mediante l'utilizzo di funzioni presenti nella libreria "Librosa", il pacchetto di Python per l'analisi di contenuti musicali. Le relative immagini degli spettrogrammi invece sono state ottenute mediante l'utilizzo della libreria "Image". Le immagini Fig. 3.1 e Fig. 3.2 riportano esempi di spettrogrammi creati in questa prima fase del programma.

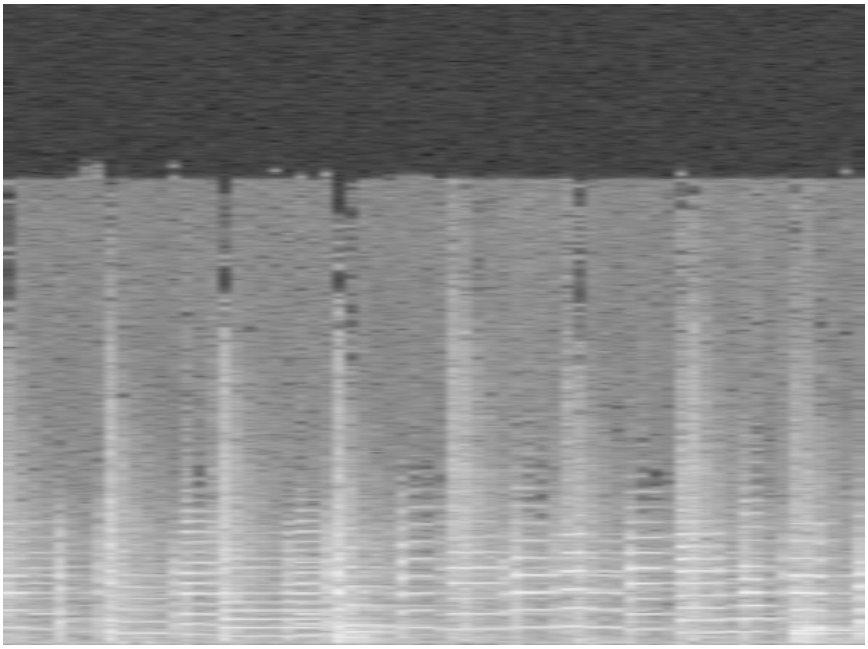


Fig. 3.1: Esempio di immagine di spettrogramma di Mel prodotto per una traccia del dataset della classe “Electric Guitar”.

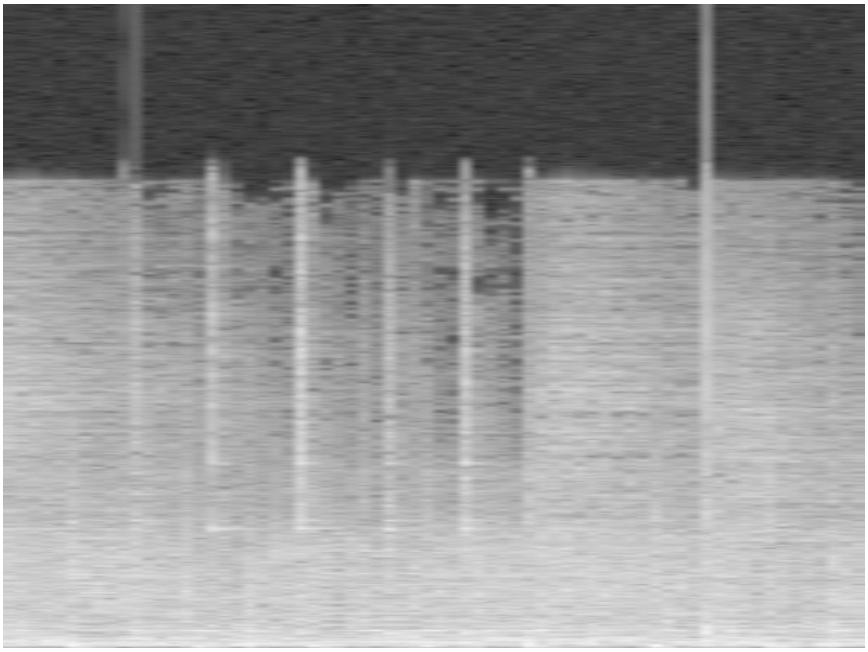


Fig. 3.2: Esempio di immagine di spettrogramma di Mel prodotto per una traccia del dataset della classe “Clarinet”.

Il passaggio successivo ha riguardato la scomposizione delle immagini in pixel, il *flattening* di ogni matrice di pixel e l'inserimento dei valori di ciascun pixel in un dataset, dove ogni riga rappresenta una traccia, e per ogni traccia viene riportata l'etichetta della classe e la lista dei pixel della relativa immagine dello spettrogramma.

Il dataset ottenuto quindi avrà dimensione 192001x6705. Prima di procedere con il pre-processing e l'addestramento dei modelli di classificazione, il dataset è stato diviso in un dataset di addestramento e un dataset di test. L'operazione è stata svolta utilizzando la funzione "train_test_split" della libreria "sklearn" di Python. Ho considerato giusto, per mantenere la proporzionalità tra le classi presente nel dataset di partenza, stratificare in base alle categorie. In questo modo nel dataset di train e di test saranno presenti le unità del dataset di partenza nella stessa proporzione. La proporzione considerata tra train e test set è 75% - 25%.

3.3 Riduzione del Dataset e addestramento dei Modelli di Classificazione

La riduzione di dimensionalità mediante PCA nei dati di addestramento e di verifica è stata ottenuta mediante l'utilizzo della funzione "PCA()", della libreria "sklearn.decomposition". La scelta del numero di componenti principali da conservare è dovuta a una valutazione basata sul *metodo del gomito*, ovvero una valutazione grafica (Fig. 3.3) sulla varianza spiegata dalle prime componenti principali. In questo caso la scelta che ha permesso di ottenere un miglior risultato è 20.

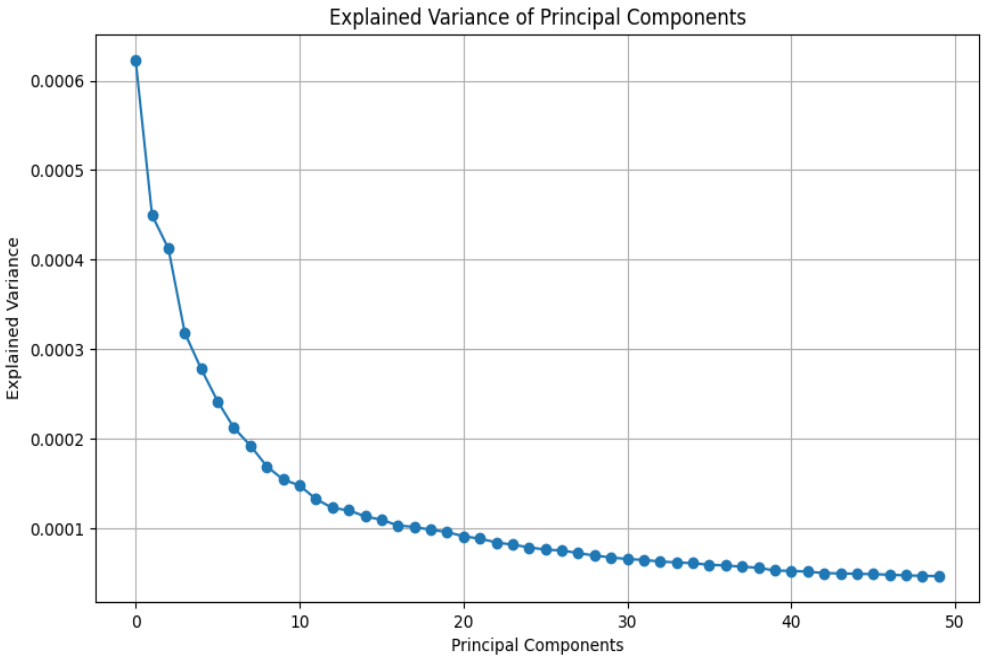


Fig 3.3: Grafico della varianza spiegata dalle prime 50 componenti principali. Si può osservare che il "punto di gomito del grafico" è situato intorno alla ventesima componente principale.

Per quanto riguarda invece i modelli di classificazione, ho utilizzato la funzione "SVD" della libreria "sklearn.svm" per implementare il modello SVM; per quanto riguarda il modello K-NN ho utilizzato la funzione "KNeighborsClassifier()" della libreria "sklearn.neighbors".

Per ottimizzare i parametri di entrambi i modelli ho considerato la funzione "GridsearchCV(). Una *grid search* è una tecnica utilizzata nel machine learning per trovare i migliori iper-parametri di un modello e consiste nel definire una griglia di possibili valori per ogni parametro e testare tutte le possibili combinazioni di valori. Per ogni combinazione di valori, un modello viene addestrato e valutato utilizzando una metrica di valutazione (F1-score). Alla fine della *grid*

search, si selezionano i valori di iper-parametri che massimizzano la metrica di valutazione.

Per quanto riguarda il modello SVM, la *grid saerch* ha permesso di capire che era necessaria un kernel non lineare, ma un kernel radiale (*Radial Basis Function*). Per quanto riguarda il modello K-NN, il parametro valutato è stato *n-neighbors*, che rappresenta il numero di vicini da considerare per la classificazione di una nuova unità. Il valore ottimo di questo parametro è risultato 5.

3.4 Metriche di valutazione dei Risultati

Una prima valutazione dei risultati globali è stata ottenuta considerando le misure di *Recall media* e *Precision media*.

3.4.1 Precision

La precisione è una misura della capacità di un modello di classificazione di classificare correttamente i campioni come appartenenti a una classe specifica tra tutte le istanze che il modello ha classificato come appartenenti a quella classe. È il rapporto tra il numero di veri positivi (TP) e la somma dei veri positivi e dei falsi positivi (FP) per una classe specifica (Fig. 3.4)

$$Precision = \frac{TP}{TP+FP}$$

La precisione fornisce informazioni sulla qualità delle predizioni del modello per una classe specifica.[8]

3.4.2 Recall

Il richiamo, noto anche come sensibilità o tasso di recupero, misura la frazione delle istanze di una classe specifica che sono state correttamente identificate dal modello. Formalmente, il richiamo per una classe specifica è calcolato come il rapporto i veri positivi (TP) e la somma tra veri positivi e falsi negativi (FN) (Fig. 3.4).

$$Recall = \frac{TP}{TP+FN}$$

Il richiamo fornisce informazioni sulla capacità del modello di identificare tutte le istanze di una classe.[8]

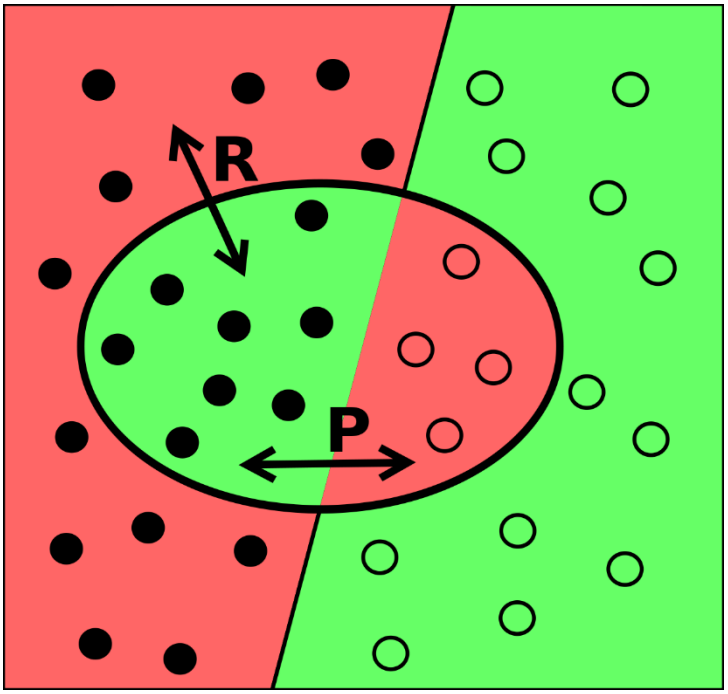


Fig 3.4: Illustrazione riassuntiva dei concetti di precision e recall.

3.5 Descrizione dei Risultati

Di seguito riporto i risultati ottenuti dalla classificazione. In particolare, discuterò prima i risultati ottenuti con il modello SVM e successivamente quelli ottenuti dal modello KNN.

3.5.1 Risultati del modello SVM

Il primo risultato che si ottiene è la tabella di contingenza tra le previsioni (colonne) e le vere classi (righe).

Strumenti	cel	cla	flu	gac	gel	org	pia	sax	tru	vio	voi
cel	20	4	5	8	9	9	5	10	3	0	0
cla	2	76	14	7	5	6	0	7	2	0	0
flu	0	6	62	7	3	6	2	3	0	0	0
gac	1	8	8	84	13	13	8	11	1	1	0
gel	5	3	3	12	83	34	20	8	7	11	1
org	3	5	6	14	23	95	15	11	9	4	4
pia	3	2	5	10	23	25	91	20	12	8	3
sax	0	5	2	9	9	5	6	72	8	5	0
tru	0	3	0	4	7	9	7	8	69	20	0
vio	0	1	1	1	6	5	6	3	10	65	18
voi	0	2	0	4	3	5	7	3	4	15	95

Per riassumere i risultati riportati nella tabella e poter valutare il modello di classificazione, ho considerato le misure di precisione e richiamo relative a ciascuna classe.

Strumento	Precision	Recall
cel	0.526	0.400
cla	0.760	0.760
flu	0.620	0.620
gac	0.628	0.840
gel	0.466	0.830
org	0.501	0.950
pia	0.544	0.820
sax	0.582	0.720
tru	0.563	0.690
vio	0.650	0.650
voi	0.791	0.863

I risultati ottenuti indicano una buona capacità del modello di discriminare le classi, in quanto i valori di precision e recall ottenuti sono significativamente distanti dal valore atteso di precision e recall in caso di classificazione casuale.

In particolare, il modello fornisce un valore di precision media del 61.7%. Una particolare difficoltà viene riscontrata nel caso della classe “Electric Guitar”, con un valore di precision del 46.6%. Questo conferma l’ipotesi iniziale di una maggiore difficoltà nel riconoscimento di strumenti non acustici.

Il valore medio di recall invece è del 72.7%. Il valore medio di recall, più alto rispetto al valore medio di precisione ottenuto, indica che il modello ha una maggiore sensibilità nel trovare gli esempi positivi per ogni classe, ma è più incline a fare predizioni positive errate (falsi positivi).

3.5.2 Risultati del modello K-NN

Allo stesso modo, il secondo modello ha prodotto la seguente tabella di contingenza.

Strumento	cel	cla	flu	gac	gel	org	pia	sax	tru	vio	voi
cel	24	13	7	11	12	11	7	6	5	1	0
cla	4	84	18	10	10	8	4	7	2	0	0
flu	2	17	67	4	5	9	4	5	0	0	0
gac	3	10	10	96	17	17	12	15	3	3	1
gel	5	5	5	16	118	29	25	12	11	10	3
org	5	7	8	18	27	107	20	15	14	6	6
pia	5	4	7	14	29	32	103	25	17	12	7
sax	1	8	4	15	16	10	11	89	13	9	2
tru	0	5	1	6	10	13	11	14	85	25	2
vio	0	2	2	2	11	9	11	6	15	80	23
voi	0	0	0	1	5	10	6	2	3	23	125

Da questa tabella ho ricavato i valori di precision e recall per ogni classe

Strumento	Precision	Recall
voi	0.742	0.691
cla	0.628	0.560
vio	0.604	0.445
org	0.545	0.639
sax	0.545	0.532
tru	0.540	0.510
flu	0.564	0.670
cel	0.500	0.750
gac	0.530	0.587
pia	0.520	0.639
gel	0.466	0.649

Come per il primo modelli, i risultati sembrano significativamente distanti dai valori attesi in caso di classificazione casuale. Per quanto riguarda la precision, si osserva un leggero peggioramento in termini di media globale rispetto ai risultati del primo modello, con una precision media del 55.3%. Si può osservare, anche in questo caso, che il valore più basso di precision lo si ottiene in corrispondenza della classe “electric guitar”. Anche per quanto riguarda il richiamo, si può osservare un peggioramento in termini di media rispetto al modello SVM, con un recall medio del 61.9%. Il valore di recall anche in questo caso è sensibilmente maggiore rispetto alla precisione.

3.6 Conclusioni

L'estrazione dei dati e la riduzione del dataset secondo i criteri illustrati hanno portato a risultati che si possono considerare buoni, ma non completamente a livello dei lavori citati nel capitolo 2. La riduzione dei dati tramite PCA ha comunque permesso di ottenere un modello meno sensibile ai fattori di disturbo e con una buona capacità di discriminare le diverse tracce audio.
Vi sono comunque diversi aspetti che offrono una possibilità di approfondimento. Per quanto riguarda l'estrazione dei dati, sempre considerando l'approccio utilizzato in questo lavoro, si potrebbero confrontare i risultati aumentando la risoluzione delle immagini, quindi la quantità di pixel presenti nel dataset. Un'altra variabile potrebbe essere quella di considerare immagini a colori, ma anche in questo caso si andrebbe ad aumentare la dimensione del dataset, rischiando di cadere in problemi di ridondanza dei dati. Per quanto riguarda invece i modelli di classificazione, si può avere un margine di miglioramento se si considerano valori diversi dei parametri; il rischio in questo caso potrebbe essere quello di ottenere modelli che si adattano in modo eccessivo al dataset, causando *overfitting*.

Bibliografia

[1] Mitsunori Ogihara. *Guest Editorial: Special Section on Music Data Mining, IEEE Transaction on Multimedia, VOL. 16, NO. 5, AUGUST 2014.*

[2] Dominika Szeligaa , Paweł Tarasiuka , Bartłomiej Stasiaka, Piotr S. Szczepaniak. *Musical Instrument Recognition with a Convolutional Neural Network and Staged Training*. 26th International Conference on Knowledge-Based and Intelligent Information & Engineering Systems (KES 2022).

[3] Alexandre M. Lucena, Caroline P. A. Moraes, Kenji Nose-Filho, Denis G. Fantinato, Aline Neves and Ricardo Suyama, *Musical Instruments Recognition using Machine Learning Techniques: MLP and SVM,* 2020.

[4] Shilpa Sonawane. *Musical Instrument Recognition using SVM,* 2018.

[5] Giorgos Mazarakis, Panagiotis Tzevelekos, and Georgios Kouroupetroglou. *Musical Instrument Recognition and Classification Using Time Encoded Signal Processing and Fast Artificial Neural Networks,* 2006.

[6] Slim Essid, Gael Richard, Bertrand David. *Musical Instrument recognition on solo performance, 2020.*

[7] S. Prabavathy, V. Rathikarani, P. Dhanalakshmi. *Musical Instrument Sound Classification Using GoogleNet with SVM and kNN Model,* 2022.

[8] Trevor Hastie, Robert Tibshirani, Jerome Friedman. *Element of Statistical Learning.* 2009.

[9] Jason Brownlee, *One-vs-Rest and One-vs-One for Multi-Class Classsification,* 2017.

[10] Bosch, J. J., Janer, J., Fuhrmann, F., & Herrera. *A Comparison of Sound Segregation Techniques for Predominant Instrument Recognition in Musical Audio Signals,* in Proc. ISMIR (pp. 559-564), 2012.