GENERI MUSICALI: STORIA E RICONOSCIMENTO TRAMITE IL MACHINE LEARNING

Baldanza Matteo[1,2], Masi Filippo[1,3], and Alessandro Zanotta[1,4]

[1]Pre-Processing
[2]Mixture Model per clustering, Model Based Distance per clustering, Classificazione con Knn e SVM
[3]Comprensione e studio delle variabili, Dbscan per clustering, Classificazione con reti neurali
[4]Clustering Gerarchico e Classificazione con Random-Forest

23 giugno 2021

Abstract

Nella generazione dei clusters, utilizzando mixture model, iterative distance-based, cluster gerarchici e dbscan, si sono ottenuti per tutti i metodi gruppi di categorie musicali significativi. I risultati conseguiti riescono a testimoniare l'originarsi di generi musicali a partire da altri e rispecchiano a pieno la musica durante il '900 e i periodi storici dei generi.

Nella parte di classificazione invece si sono testati metodi quali k-nearest neighbors, support vector machine, random forest e reti neurali. Si è riuscito a ottenere un'ottima performance tramite quest'ultimo ottenendo un accuracy pari a 0.72. Si riesce quindi correttamente ad identificare un genere musicale con una probabilità del 72%. Il valore è molto buono in quanto, essendo 10 generi, la probabilità di indovinare uno casualmente è del 10%

1 Intoduzione

Il dataset di riferimento utilizzato per le nostre analisi è fruibile al seguente link `https://www.kaggle.com/harish24/music-genre-classification`. E' composto da 1000 istanze e da 28 features. Ogni istanza si riferisce a una traccia musicale della lunghezza di circa 30 secondi. Le tracce dopo essere state registrate, sono state sottoposte a uno strumento di rilevazione delle frequenze sonore. Le frequenze rappresentano la base per il calcolo delle variabili create nel campione. Il lavoro si basa su due differenti task 1.1.

1.1 Obbiettivi

- La Cluster analysis ha come fine la creazione di gruppi che siano il più possibili distinti tra di loro (eterogenei) e concordi al loro interno (omogenei) in maniera tale che siano molto diversi tra di loro. E' di interesse nel lavoro qui svolto andare ad indagare come l'unione di diverse tracce musicali venga effettuata in base alle carattestiche dei suoni riprodotti. In particolare si vuole capire se i metodi usati riescano a mostrare le similitudini tra generi musicali affini in senso ritmico/melodico e che condividono lo stesso periodo storico di nascita.
  Nel corso del '900 infatti si sono sviluppati la maggior parte dei generi nel dataset qui presente e spesso la musica ha rappresentato le caratteristiche del periodo storico. Parte dei generi hanno origine proprio da altri, a formare quasi uno sviluppo a *matrioska* (come ad esempio il blues che nasce dal jazz, dà vita poi al rock che all'estremo si può trasformare nel metal); oppure alcuni nascono da molteplici (come l'hip hop che si origina dal pop, dalla disco e dal raggae).

Il fine ultimo è quindi di capire tramite clustering se si riescono a estrapolare da delle semplici tracce questi tipi di sviluppi musicali.
- La fase di classificazione delle istanze mira invece a trovare il miglior metodo/modello capace di identificare un genere musicale (tra i 10 presenti) sulla base di una traccia inputata negli algoritmi, tramite i valori delle variabili presenti nel dataset. Si desidera quindi addestrare la macchina informatica (il computer) affinché riesca a imparare dai dati forniti ad assegnare delle previsioni per ogni melodia musicale ricevuta come input.
  Metodi di Machine Lerning e Statistical learning di questo tipo possono tornare utile per siti, applicazioni e piattaforme musicali che hanno un bacino di brani musicali molto elevato e devono agevolare l'organizzazione di essi classificandoli nei vari generi di appartenenza.

2 Pre-Processing

2.1 Le variabili

Come già anticipato nella sezione 1 , il dataset contiene mille registrazioni di tracce musicali; per analizzarle e poter cogliere le particolarità che meglio le riescono a contraddistinguere, sono state fornite ventisei *features* che rappresenteranno le variabili dei modelli poi illustrati durante l'elaborato. Ognuna di esse ha il merito di riconoscere i segnali audio e di estrarre un valore per ogni traccia che riassuma la proprietà rappresentate. E' bene conoscere e comprendere prima di procedere oltre il significato delle features presenti.

Per ottenere le istanze del dataset, le tracce musicali sono state raffiguarate su uno spettogramma che è la rappresentazione grafica dell'intensità di un suono in funzione del tempo e della trasformazione logaritmica della frequenza. Sull'asse delle ascisse è riportato il tempo in scala lineare;sull'asse delle ordinate è riportata la frequenza in scala lineare o logaritmica; a ciascun punto di una data ascissa e ordinata è assegnata una tonalità di colore, rappresentante l'intensità del suono in un dato istante di tempo e a una data frequenza. Uno spettrogramma si ottiene, di solito, suddividendo l'intervallo di tempo totale (cioè quello relativo all'intera forma d'onda da analizzare) in sottointervalli uguali (detti finestre temporali) di durata da 5 a 10 ms e calcolando la trasformata di Fourier della parte di forma d'onda contenuta in ciascuna finestra, che fornisce l'intensità del suono in funzione della frequenza. Le trasformate di Fourier, relative alle diverse finestre temporali, vengono poi assemblate in base alle diverse note musicali. Dallo spettrogramma della traccia sono state calcolate le variabili presenti in questo dataset, in particolare:

- `"chromaStft"`: valore che riassume le frequenze emesse dalla traccia, trasformate secondo Fourier e infine rappresentate con tonalità di colori presenti all'interno dello spettrogramma.
- `"rmse"` : indica la radice quadrata dell'errore quadratico medio (*root mean square error*) tra i valori delle frequenze ottenute nel riconoscimento del suono, e i valori usati come stime all'interno dello spettro. Dipende chiaramente dal livello della risoluzione della traccia.
- `"spectralCentroid"`: indica dove si trova il centro di massa dello spettro. Percettivamente, ha una solida connessione con la brillantezza del suono.
- `"spectralBandwidth"`: la larghezza di banda dell'onda a metà del picco massimo. La larghezza di banda spettrale di uno spettrofotometro è correlata alla risoluzione del suono.
- `"rolloff"`:
- `"zeroCrossingRate"`: la velocità con cui un segnale nello spettrogramma passa da positivo a negativo e viceversa. Questo valore rappresenta una caratteristica chiave per classificare i suoni di percussioni e per riconoscere la presenza o meno di voce umana.
- `"mfcc"`: 20 variabili che corrispondono ai coefficienti cepestrali di frequenze mel. Indicano la velocità del cambiamento del contenuto cepestrale di un segnale audio. Il cepstrum è ottenuto convertendo le frequenze dello spettrogramma da Hz nella scala Mel (che simula in maniera più precisa la percezione umana del suono), attraverso una traformata logaritmica. Evidenziano la differenza di dissipazione di un suono che chiaramente dipende da volume, timbro e acutezza. Riescono quindi a distinguere due segnali con stessa frequenza

ma di diverso suono, simulando la percezione umana al meglio.

Oltre a queste, nell'insieme di dati sono presenti la colonna relativa ai nomi delle variabili e la colonna `"label"` che mostra l'etichetta del genere musicale corretto.

2.2 Analisi delle variabili

La variabile in posizione uno, contenente l'identità della traccia musicale (es. "*blues0001*") è stata eliminata in quanto superflua per il lavoro.

La nostra variabile target è `"label"`. La prima cosa da visonare è la frequenza con cui ogni categoria appare all'interno del nostro dataset. Il risultato è il seguente:

Blues	Classical	Country	Disco	Hip Hop
100	100	100	100	100
Jazz	Metal	Pop	Reggae	Rock
100	100	100	100	100

Le categorie sono ben bilanciate, il che facilita il nostro lavoro in quanto non c'è bisogno di riequilibrare le classi. Inoltre il dataset in questione non ha missing values.
Per la nostra analisi, considerato che si andranno ad utilizzare metodi di clusterng basati sulle distanze, è importante anche valutare metodi come standardizzazone e/o normalizzazione. Le variabili in questione hanno range molto diversi il che porterebbe a pesi differenti nei vari algoritmi. Si riportano qui i grafici delle variabili normalizzate per avere anche un'idea dei vari outliers presenti all'interno delle variabili.

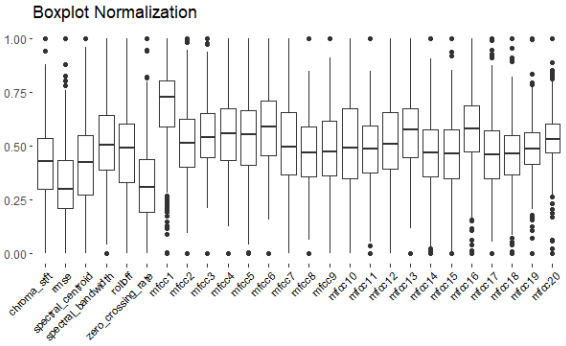

Figura 1: Dati Normalizzati

In figura 1 vengono illustrate le variabili a confronto che appartengono ad uno stesso intervallo e si può osservare come le varie medie siano diverse.

E' da notare la presenza di alcuni outliers in quasi tutte le variabili (si fa notare che senza la normalizzaione o standardizzazione non c'era evidenza di tali valori). In `mcff1` si nota la presenza di molteplici punti anomali e si è cosi deciso, anche perché tale variabile risulta molto importante nelle analisi successive (si vedrà nel 2.3 in particolar modo), di indagare questi valori. Il risultato è davvero di particolare interesse. Tutti gli outliers (poco più di una ventina) si riferiscono alla `label` classical. La variabile `mfcc1` contiene i coefficienti cepestrali di maggiore importanza per ogni registrazione musicale. Questi valori si riferiscono alla velocità del cambiamento del contenuto cepestrale di un segnale audio che varia in maniera significativa se il volume è minore. Effettivamente si nota come le tracce anomale hanno un audio spesso molto basso, con pause frequenti e durevoli che rallentano la velocità del cambiamento del cepstrum e quindi rendono il valore del mfcc1 molto più basso.

Si è deciso quindi di mantenere all'interno del clustering queste anomalie in quanto riflettono appieno le caratteristiche di una classe (quella della `label` classica) e per la stessa ragione sono state tenute anche per la fase di classificazione.

2.3 Riduzione dimensionalità tramite RandomForest

Data un'elevata quantità di variabili, si è deciso di utilizzare un metodo basato su RandomForest per selezionare le features che sono state usate per tutto il resto del lavoro. In breve la selezione eseguita si basa sulla valutazione della misura dell'impurità. Quando si addestra un albero, è possibile calcolare quanto ogni variabile riduce questo valore e aumenta il guadagno di informazione. Più una feature riduce l'impurità, più importante è tale variabile. Nel random forest, la diminuzione dell'impurità può essere usata tra gli alberi per determinare la rilevanza finale delle variabili (le variabili selezionate nella parte superiore degli alberi sono in generale più importanti di quelle selezionate nei nodi finali degli alberi, poiché solitamente le divisioni supe-

riori portano a maggiori guadagni di informazioni). Il risultato è il seguente:

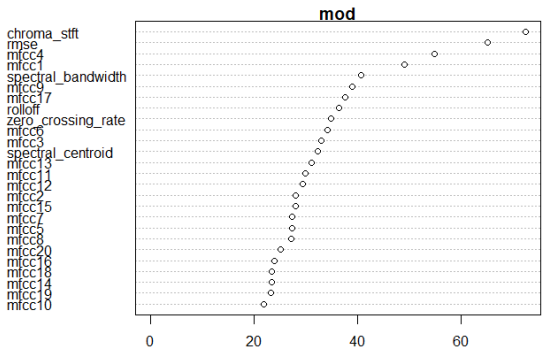

Figura 2: Scelta delle variabili-Impurità di Gini

Si può osservare che la variabile più influente all'interno del dataset è `chromastft` mentre la seconda è `rmse`, seguono poi `mfcc4` e `mfcc1`. Il motivo per il quale abbiamo scelto in precedenza di analizzare i valori anomali di `mfcc1` risulta ora molto chiaro. Si procede scegliendo le prime 10 variabili in quanto poi non c'è evidenza netta di un miglioramento profondo.

3 Clustering

3.1 Mixture Model

Il primo approccio provato è basato sull'utilizzo di mixture model. Considerato il metodo di tipo probabilistico che utilizza la massima verosimiglianza come condizione per l'adattamento del modello, non è necessario ridimensionare i dati. Se una variabile ha una varianza maggiore di un'altra, la procedura di ottimizzazione sarà in grado di apprenderla e adattare le varianze (o le matrici di covarianza nel caso multivariato) di conseguenza. Si è utilizzato un approccio differente a quello classico non basato solamente sull'utilizzo dell'algoritmo EM ma anche sulla scomposizione della matrice di varianze e covarianza. I cluster sono ellissoidali, centrati sul vettore delle medie μ_k con caratteristiche geometriche quali volume, forma e orientamento, determinate dalla matrice di covarianza Σ_k. La scomposizione della matrice di covarianza può essere ottenuta mediante

$$\Sigma_k = \lambda_k D_k A_k$$

ove in ordine si ha uno scalare che controlla il volume dell'ellissoide, una matrice diagonale che specifica la forma dei contorni di densità e una matrice ortogonale che determina l'orientamento dell'ellissoide corrispondente.
Il numero di componenti e il tipo di parametrizzazione della covarianza vengono selezionati utilizzando il criterio di informazione bayesiano (BIC), il confronto ottenuto è il seguente:

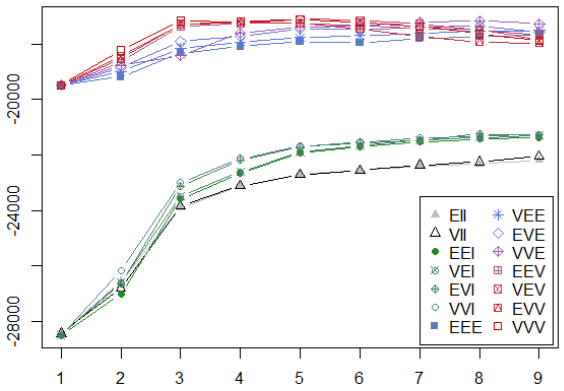

Figura 3: Analisi del BIC

Sebbene il BIC è massimizzato per cinque gruppi, si può notare dal grafico 3 come in realtà il valore con k=4 (numero di gruppi), sia molto similare. Per il nostro obbiettivo iniziale si è scelta una modelizzazione con 4 gruppi per maggior interpretazione. Il risultato del clustering, avendo a disposizione le etichette è il seguente:

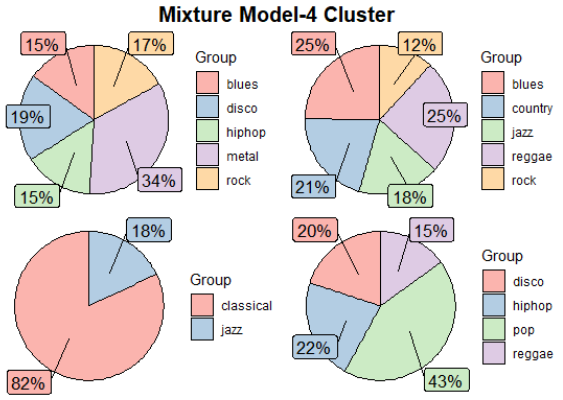

Figura 4: Risultato Clustering con Mixture Models

Per le interpretazione dei risultati ci sarà spazio

dopo aver visto l'utilizzo di tutti i metodi in 3.5. Per ora si ribadisce il fatto che essendo l'obbiettivo quello di capire se esistono legami tra le tracce e come storicamente è nata e cresciuta la musica, è necessario porre attenzione alle percentuali di generi presenti in ogni gruppo.

3.2 Iterative Distance-Based

Si è optato per algoritmi quali KMeans e KMedians i quali utilizzano come misura di distanza quella Euclidea e quella di Manhattan. I dati sono stati standardizzati in quanto algoritmi basati su misure di distanza.
I risultati sono stati confrontati sulla base di un criterio interno ovvero la Silhouette, che è stata massimizzata in entrambi i casi valutando tutti i possibili iperparametri (iperparametro in questione è k cioè il numero dei gruppi). Il massimo valore della silhouette media ottenuto è di circa 0.50, il cui significato ci indica che la partizione ottenuta è attendibile. I valori della silhouette con entrambi i metodi sono praticamente identici. Si è preferito quindi tenere come risultato finale quello del KMeans in quanto la distanza euclidea è più semplice e veloce da calcolare computazionalmente rispetto a quella di Manhattan. Selezionato il valore k=3, il risultato della partizione dei gruppi è stato il seguente:

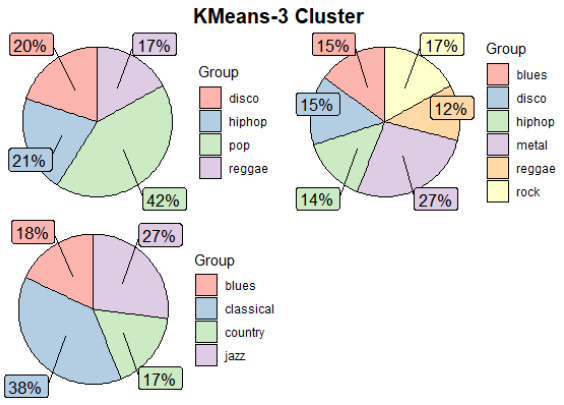

Figura 5: Risultato KMeans

Anche in questo caso i risultati ottenuti sono interessanti dal punto di vista storico-musicale.

3.3 Cluster Gerarchici

Il secondo metodo utilizzato è stato mediante i clustering gerarchici di tipo agglomerativo. In questo caso si sono utilizzati i dati grezzi e non i dati standardizzati o normalizzati. Il migliore risultato è stato ottenuto con il metodo per il calcolo della distanza tra i clusters denominato "average" ovvero calcolando la distanza media tra tutte le possibili coppie che compongono i due gruppi in esame. Si è ottenuto infatti un valore della Silhouette media pari a 0.54, numero che assicura l'attendibilità dei gruppi trovati. Per il calcolo della matrice delle distanze è stata utilizzata la distanza euclidea. Il taglio è stato effettuato per tre gruppi:

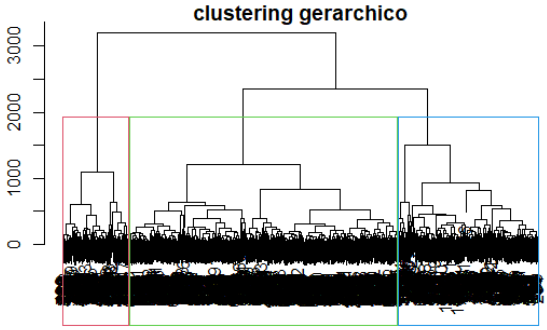

Figura 6: Clustering Gerarchico

Si nota come i tre clusters identificati siano ben riconoscibili, il che implica maggior similarità all'interno dei gruppi trovati e diversità tra di essi. Si propone anche un grafico basato sulle due variabili "più importanti" (si guardi 2.3) per enfatizzare tale suddivisione.

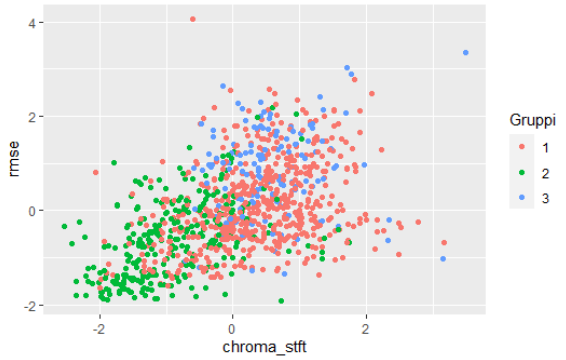

Figura 7: Cluster Sulla base di due variabili

E' possibile notare come all'aumentare delle prime due variabili per importanza, i cluster vengano sempre più ben delineati, a tal proposito viene riportato il grafico 8 a torta con la composizione dei vari gruppi.

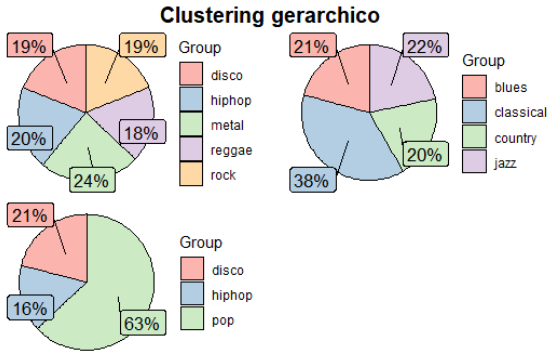

Figura 8: Risultato Cluster gerarchici

Per le interpretazioni si rimanda ancora una volta a 3.5.

3.4 DbScan

L'ultimo tra gli algoritmi sperimentati per la creazione di clusters è il metodo di raggruppamento basato sul concetto di densità delle istanze nello spazio, ovvero il DbScan (*Density-based spatial clustering of applications with noise*). Questo metodo prevede la divisione di punti nello spazio in:

- Core points
- Border points
- Noise points,

sulla base del numero di punti presenti in una sfera di raggio ϵ maggiore o meno di una soglia τ (con $\tau, \epsilon \in \mathbf{N}$).

I due parametri sono da inserire come input dell'algoritmo. Le previsioni fornite variano a seconda dei due valori, per questo motivo si è proceduto sviluppando una funzione che permettesse di ripetere l'algoritmo al variare dei due parametri in un range prestabilito (Automated Machine Learning). Per ogni ripetizione è stata analizzata la performance dei clusters ottenuti, attraverso il calcolo della *silhoutte*, e da questa serie di valori è stata individuata la coppia di parametri che ha massimizzato la quantità:

	Value
Silhouette	0.46
Epsilon	0.21
Soglia	33

Si sono utilizzati dati normalizzati che conservano le differenze di medie e possiedono lo stesso range.

Dopo aver trovato i parametri *ottimi* da inputare, si è proceduto sviluppando il raggruppamento. Si è ottenuto il seguente risultato in fig 9

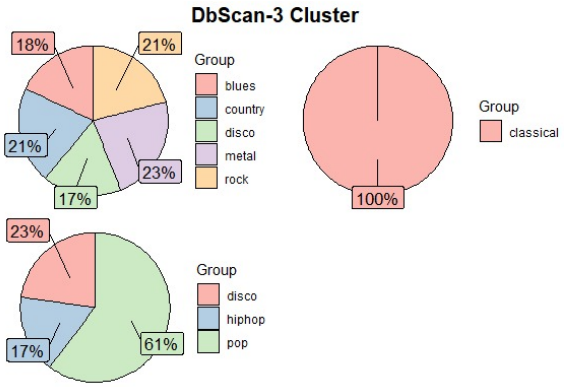

Figura 9: Dbscan clusters

Si passa ora finalmente all'analisi dei risultati ottenuti in tutti i metodi.

3.5 Interpretazioni clustering

Alla luce degli obiettivi fissati prima dell'inizio dell'analisi di clustering, si può constatare come i gruppi conseguiti mostrino un'ottima eterogeneità e in alcuni dei casi anche una discreta omogeneità al loro interno.

I risultati dei dbScan e del metodo basato sulle misture rivelano, in maniera netta, la differenza della musica classica rispetto agli altri generi: il primo algoritmo dedica un gruppo unicamente ad essa con un'eterogeneità al suo interno nulla; mentre nell'altro approccio si nota sempre una presenza maggioritaria in un unico gruppo, che viene in parte condiviso con il jazz. Questo notevole esito è sicuramente la prova della natura differente dei suoni e dei ritmi della musica classica, della sua diversa strumentazione (tipica dell'orchestra) e del suo diverso periodo storico di spicco ('700 e '800 e primi del '900) rispetto agli altri generi musicali. Inoltre viene illustrato come l'unico tra i generi ad avvicinarsi è il jazz. A documentare la vicinanza dei due generi si sottolinea la nascita del jazz anticipata rispetto agli altri (primi del '900), passaggi musicali e giri di accordi più arguti in questi generi, e la consuetudine di non usare frequentemente il canto umano.

Per quanto riguarda invece gli altri gruppi, in tutti gli approcci verificati, si nota come si ha una buona divisione tra i generi musicali nati in annate meno moderne:`jazz`, `blues`, `country`, `metal` e `rock` e `classical` e i generi musicali nati e rimasti più in voga in periodi storici vicini ai giorni nostri : `pop`, `disco` e `hip hop`.

Tra i generi più *all'antica* in alcuni casi si è sviluppata la distinzione tra quelli caratteristici di suoni distorti, elettrici e volumi aggressivi, rock e metal, e il resto (come nella sezione3.1 dove in un gruppo i due generi misurano insieme il 51% e in kmeans e gerarchici che occupano intorno al 45%).

Un genere musicale che è stato spesso inserito in due clusters diversi, è la `disco`. L'inserimento nello stesso cluster di rock e blues, (si veda 5 e 9) trova conferma nel fatto che il genere musicale abbia periodo storico di nascita e maggiore tendenza durante la seconda metà degli anni 70' e della prima metà degli anni 80' (periodo tipico dei "Bee Gees" con la colonna sonora del film "la febbre del sabato sera"), e quindi alcune tracce hanno suoni tipici del periodo. Tuttavia è molto variato il concetto di canzone disco (canzone tipica delle discoteche) nel corso della storia e si è avvicinato di più alla musica pop/hiphop con l'uso di suoni più elettronici. Alcune delle tracce contenenti suoni più moderni sono così raggruppate negli altri gruppi.

Un altro genere musicale che non viene facilmente riunito in un unico gruppo è il raggae. Questa tipo di musica nasce durante la fine degli anni sessanta e quindi conserva alcune caratteristiche del Rock e Country ma contemporaneamente dà vita a ritmi dall'andamento più spezzato con forte risalto ai bassi dando origine ai gruppi più innovativi e moderni (hip hop e disco).

L'origine del genere hip hop si nota in maniera evidente in quasi tutti i metodi usati, infatti viene quasi sempre raggruppato con pop, disco e raggae.

4 Classificazione

Si è deciso di dividere il dataset in training e test set. Tutti i metodi che si illustreranno in questa sezione sono stati addestrati nel training set tramite Repeated-Cross-Validation. In questo modo si sono ricercati gli iperparametri in grado di minimizzare il generalized-error.

4.1 K-nearest neighbors

Il metodo in questione ha un solo iperparametro. L'obbiettivo è quello di massimizzare l'accuracy per classificare al meglio il genere musicale delle tracce. Il valore che si è ottenuto per minimizzare il generalized-error (valore ottenuto 0.4134236) è di k=0.59. Sebbene il valore sembra alto in realtà è molto buono in quanto si classifica con una percentuale di correttezza del 60%. Per verificare la robustezza del risultato si è calcolato l'errore empirico pari a 0.5358974. L'accuratezza in questo caso è più bassa ma comunque si tende ad accettare il valore di k ottenuto in precedenza in quanto non ci si aspetta un grossa distorsione da questi valori. Si ritiene comunque che questo tipo di approccio sia troppo semplicistico per il caso in questione.
Nel test quello che si è ottenuto è il seguente risultato:

Accuracy: 0.565

Sensitivity:				
blues	classical	country	disco	hiphop
0.50	0.85	0.55	0.40	0.30
jazz	metal	pop	reggae	rock
0.60	0.85	0.75	0.60	0.25

Innanzitutto si nota che l'accuracy è in linea coi livelli delle performance effettuate nel training set. Si è deciso di riportare anche i valori della sensitivity (misura che indica i corretti classificati rispetto al totale delle unità da predire). Si può facilmente vedere come la classe che questo metodo fa fatica a predire sia quella della musica Rock seguita dalla musica hip hop. I generi musicali vengono confusi dall'algoritmo forse per la troppa somiglianza tra le tracce, ad esempio la musica hip hop potrebbe essere facilmente scambiate con la musica disco e/o pop.
La classe invece in cui vengono classificate correttamente più istanze, come facilmente prevedibile dalle analisi passate, è quella della musica classica a pari con la musica metal.

4.2 Support Vector Machine

Si è scartata subito l'opzione di utilizzare una SVM del tipo lineare in quanto i nostri dati difficilmente saranno separibili linearmente essendo complicati. Si è optato quindi per un confronto tra support vector machine non lineare a kernel "*radial*" e a kernel "*polinomiale*". Il confronto come nel caso precedente è stato effettuato sul training tramite repeated cross validation. Gli iperparametri in questione da scegliere sono *C* e *sigma*. *C* è il parametro di penalità che indica all'algoritmo quanto errore massimo è ammesso. In questo modo è possibile controllare il compromesso tra limite decisionale e l'errore di classificazione. *Sigma* definisce quanta influenza ci deve essere tra i punti e la linea di separazione, quando è più alto il suo valore, i punti vicini avranno un'influenza elevata viceversa quando è bassa anche i punti lontani contribuiranno al fine di ottenere il confine di decisione.

I risultati migliori sono stati trovati con l'utilizzo di una kernel di tipo radiale. Gli iperparametri trovati non sono quelli che massimizzano l'accuracy del training set in quanto tali valori portavano a problemi di overfitting nella verifica dei risultati sul training set. Si è scelta quindi una combinazione di parametri tale per cui i due errori coincidessero (vicini fra di loro) per essere sicuri di evitare over/under performance nel set di dati non a disposizione. I valori usati sono i seguenti:

	Value
C	3.5
Sigma	0.1

Si riportano ora gli errori individuati nelle varie fasi:

	Value
Generalized-Error	0.364375
Empirical-Error	0.2012
Test-Error	0.365

L'errore nel test set è quello che ci si aspettava dopo la valutazione del generalized error. L'accuracy quindi è più elevata rispetto al metodo Knn utilizzato in precedenza. Si riporta anche la tabella con i valori della sensitivity:

Accuracy: 0.635

Sensitivity:				
blues	classical	country	disco	hiphop
0.60	0.90	0.55	0.55	0.45
jazz	metal	pop	reggae	rock
0.70	0.95	0.80	0.60	0.25

Anche qui il rock è il più complicato da classificare tra i vari generi. La musica classica e la musica metal rimangono le più semplici da identificare tramite questo algoritmo. Il risultato ottenuto è molto buono, un accuracy pari a 0.635 per il nostro problema è un ottimo risultato. Rispetto al knn la musica hip hop viene riconosciuta maggiormente.

4.3 Random Forest

Il terzo metodo di classificazione implementato è stato il random forest, sono stati conseguiti ottimi risultati in termini di robustezza, in quanto l'errore empirico e l'errore generalizzato hanno assunto valori piuttosto simili, 0.375 il primo e 0.3875 il secondo. Il metodo Random forest possiede due iperparametri, il numero di sottovariabili da campionare (indicato con m) e il numero di alberi che

compongono la foresta. I migliori esiti sono stati ottenuti con un numero di variabili da campionare pari a 7 e un numero di alberi pari a 2500. I risultati ottenuti nel test set (accuracy e sensitivity) sono riportati nella tabella sottostante.

Accuracy: 0.65

Sensitivity:				
blues	classical	country	disco	hiphop
0.70	0.95	0.75	0.65	0.55
jazz	metal	pop	reggae	rock
0.70	0.85	0.75	0.45	0.15

Come ulteriore testimonianza della robustezza delle analisi fin qui effettuate l'accuracy sul test set si avvicina ai valori ottenuti precedentemente tramite cross validation. Un altro valore riportato in tabella è la sensitivity, è interessante notare come anche in questo caso, la classe con la sensitiviy più bassa risulta essere la classe relativa al genere Rock. Una possibile spiegazione teorica oltre che storica per tale risultato è che il genere Rock racchiude al suo interno la storia della musica, risultando quindi formato da un mix di componenti difficili da classificare.

4.4 Reti Neurali

Le reti neurali sono uno degli metodi più di rilievo del machine learning. Come ultimo approccio di classificazione si è proceduto con la creazione di questi algoritmi sfruttando la loro vera potenzialità: formare una struttura computazionale densa di connessioni tra i neuroni, che utilizzi come attivazione funzioni non lineari. A differenza dei precedenti procedimenti è stato utilizzato tutto il dataset in quanto si è voluto addestrare la rete su tutte le informazioni in nostro possesso. Come nei casi precedenti si è effettuato sul training un repeated cross validation in modo tale da rintracciare la dimensione ottima dello strato nascosto che rendesse massima l'accuracy sul Validation Set. La rete neurale costruita ha un unico Hidden Layer formato da 10 neuroni con un tasso di decadimento di 0.44 con i seguenti esiti:

	Value
Generalized-Error	0.369
Empirical-Error	0.233
Test-Error	0.276

Si riportano ora i valori della sensitivity calcolati sul test-set.

Accuracy: 0.724

Sensitivity:				
blues	classical	country	disco	hiphop
0.64	0.96	0.68	0.80	0.64
jazz	metal	pop	reggae	rock
0.84	0.88	0.80	0.60	0.40

Si può notare come la sensitivity su ogni classe migliori rispetto a tutti i metodi precedenti. Il miglior guadagno lo si ha per la classe rock la classe più difficoltosa da riconoscere. Le reti neurali riescono infatti a riconoscere meglio tale genere musicale così come l'hip hop.

Dati i risultati ottimi si riporta in questo caso la confusion matrix del test set (figura 10)

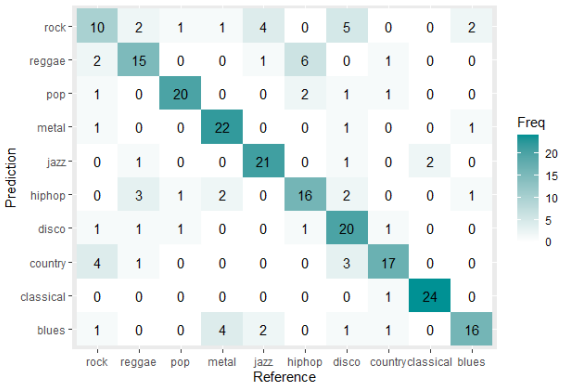

Figura 10: Matrice di Confusione Reti Neurali

Conclusioni

Gli algoritmi utilizzati nel clustering hanno evidenziato le similitudini tra i vari generi in base all'epoca in qui sono nati. Inoltre si è testimoniato l'originarsi di generi musicali a partire da altri.

I risultati ottenuti nella classificazione hanno evidenziato una precisione massima del 72% (con reti

neurali) nel riconoscere un genere musicale. Il genere rock e il genere hip hop sono risultati i più complicati da etichettare forse per la loro similitudine in termini di melodie, ritmo e strumentazione con metal,pop e disco rispettivamente.

Si noti che provando casualmente a indovinare un genere musicale la probabilità di prevederlo correttamente è solo di 0.1; per questo motivo l'utilizzo del machine learning nel seguente ambito si è rivelato di grande efficacia e può essere ottimo aiuto per l'organizzazione di brani musicali sulle piattaforme in streaming (per esempio Spotify) e per applicazioni atte al riconoscimento audio (per esempio Shazam)

Bibliogragia e Sitografia

1 Foundation of Machine Learning-second edition, Mehryar Mohri
2 https://elearning.unimib.it/course/view.php?id=31159
3 https://data-flair.training/blogs/python-project-music-genre-classification/
4 http://marsyas.info/downloads/datasets.html
5 https://en.wikipedia.org/wiki/Spectrogram
6 https://en.wikipedia.org/wiki/Spectral_centroid
7 https://www.sciencedirect.com/topics/engineering/zero-crossing-rate#:~:text=4.3.,the%20length%20of%20the%20frame
8 https://www.audiolabs-erlangen.de/content/05-fau/professor/00-mueller/02-teaching/2016s_apl/LabCourse_STFT.pdf