



Corso Data Italy

Report Unsupervised Learning

Generi musicali in base alle frequenze? Un annoso dilemma

Studenti:

Francesco Turilli

Sabino Acquaviva

Angelo Imperiale

Nicolò Pace

Massimiliano Mutzu Martis

ANNO CORSO 2021

Indice

Abstract	1
1 Descrizione Dataset	2
2 Clustering Analysis	4
3 Principal Component Analysis (PCA)	9
Conclusioni	12

Abstract

Un genere musicale è una categoria convenzionale che identifica pezzi di musica come appartenenti a una tradizione condivisa o a un insieme di convenzioni. Va distinto dalla forma musicale e dallo stile musicale. Le caratteristiche estratte da queste onde possono aiutare la macchina a distinguerle. In questo report vogliamo analizzare se sia possibile questa classificazione e capire se una canzone appartiene a un determinato genere musicale utilizzando metodi oggettivi e non soggettivi, come ad esempio è stato fatto in questo lavoro utilizzando una tecnica di clustering non supervisionato sull'analisi di uno spettro di frequenze e numero di battiti effettuato su un dataset di 1000 sample.

I dati sono stati ottenuti campionando 30 secondi di ogni sample con una decomposizione dello spettro in frequenza, a cui successivamente è stato aggiunta una colonna per descrivere il genere di appartenenza.

I metodi utilizzati sono stati: clustering con K-Means (per la robustezza si è utilizzato successivamente il trimmed k-means), Partitioning Around Medoids e dato l'elevato numero di predittori si è esplorato un metodo di riduzione dimensionale (Principal Component Analysis).

1. Descrizione Dataset

Le caratteristiche di questo dataset sono estratte dal set di dati utilizzato che consiste in 1000 tracce audio di 30 secondi ciascuna. Contiene 10 generi, ognuno rappresentato da 100 tracce. Le tracce sono tutti file audio 22050Hz Mono 16-bit in formato .wav. Le caratteristiche sono estratte usando la libreria libROSA di Python, settata per analisi di musica e audio.

Credits del dataset utilizzato: sito web marsyas.info.

Ora di seguito una breve descrizione delle feature:

- **Tempo**: la velocità con cui viene suonato un passaggio di musica;
- **Beats**: Unità ritmica nella musica;
- **Chroma_stft**: Trasformata di Fourier nel breve periodo di tempo;
- **Rmse**: Radice errore quadratico medio;
- **Spectral_centroid**: Indica dove si trova il "centro di massa" dello spettro;
- **Spectral_bandwidth**: È l'intervallo di lunghezza d'onda in cui una quantità spettrale irradiata non è inferiore alla metà del suo valore massimo;
- **Rolloff**: la ripidità di una funzione di trasmissione con la frequenza;
- **Zero_crossing_rate**: La velocità con cui il segnale passa da positivo a negativo o viceversa;
- **Mfcc**: Coefficienti ricavati dall'analisi di Fourier dello spettro di potenza.

Inizialmente nel repository di kaggle (<https://www.kaggle.com/insiyeeah/musicfeatures>) abbiamo preso in esame due dataset che pensavamo complementari, uno da 1000 record e l'altro da 200. Abbiamo unito i due dataset ma attraverso l'analisi preliminare, ci siamo accorti che le osservazioni presenti nel secondo dataset erano una semplice ripetizione di record del primo.

La prima analisi dunque è costituita dall'eliminazione dei record duplicati e dalla successiva ricerca di eventuali valori NA e Null e abbiamo constatato l'assenza di entrambi al suo interno.

Una volta ripulito il dataset si è passato all'analisi operativa vera e propria. In primo luogo abbiamo ricercato la presenza di valori uguali a 0 per poter scegliere la migliore distanza per le successive funzioni di clusterizzazione.

Successivamente, vista la presenza di feature con ordini di grandezza differenti, si è scelto di standardizzare valori, nella fattispecie, per esempio per le feature RMSE e spectral_centroid.

Per effettuare un'analisi visiva del dataset, abbiamo scelto di plottare i valori della correlazione tra le feature all'interno di un pairsplot.

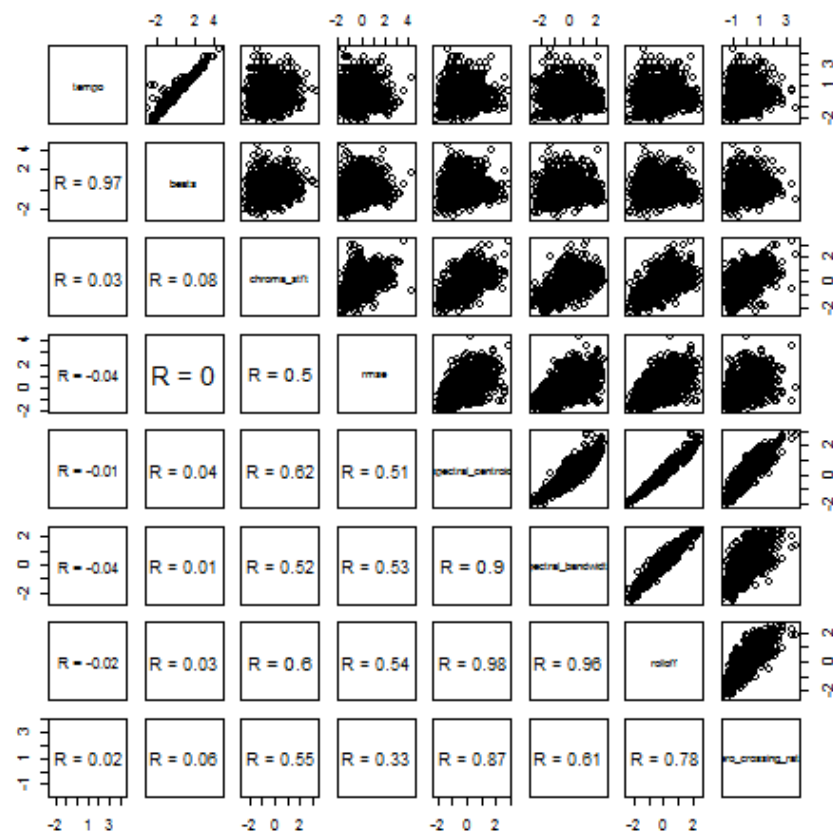


Figura 1.1: Esempio di pairsplot dalla prima all'ottava variabile

2. Clustering Analysis

In questa fase ci siamo dedicati alla clustering analysis.

Per prima cosa ci siamo chiesti in quanti cluster era opportuno suddividere il nostro data-frame, quindi a tal fine abbiamo utilizzato la funzione `fviz_nbclust` per scoprire il numero ottimale di raggruppamenti, ottenendo come risultato ottimale tre cluster.

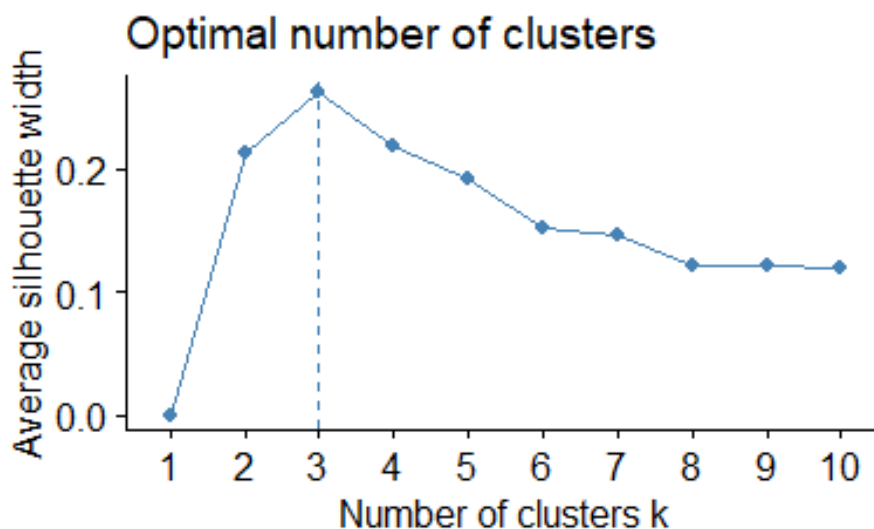


Figura 2.1: Plot della silhouette della k-means

Utilizzando il risultato ottenuto precedentemente, siamo in grado di clusterizzare il data-frame utilizzando la metodologia K-Means, al fine di trovare dei raggruppamenti attorno a dei centroidi.

	1	2	3
blues	0	47	53
classical	1	4	95
country	24	26	50
disco	45	53	2
hiphop	53	42	5
jazz	23	13	64
metal	3	97	0
pop	93	0	7
reggae	46	26	28
rock	29	52	19

Figura 2.2: Table dei Cluster

Come si evince dalla tabella precedente possiamo notare che l'algoritmo K-Means ha messo in evidenza come la quasi totalità delle tracce pop appartengono al primo cluster. Il secondo è fortemente contrassegnato dalle tracce metal; infine il terzo cluster dalla musica classica. Tutti gli altri generi sono distribuiti tra questi tre cluster denotando come vengano influenzati dai tre generi preponderanti.

Visto il grafico della silhouette, per cercare riscontri relativi ai risultati appena ottenuti abbiamo applicato la PAM (Partitioning Around Medoids) al dataset, ottenendo ciò che segue:

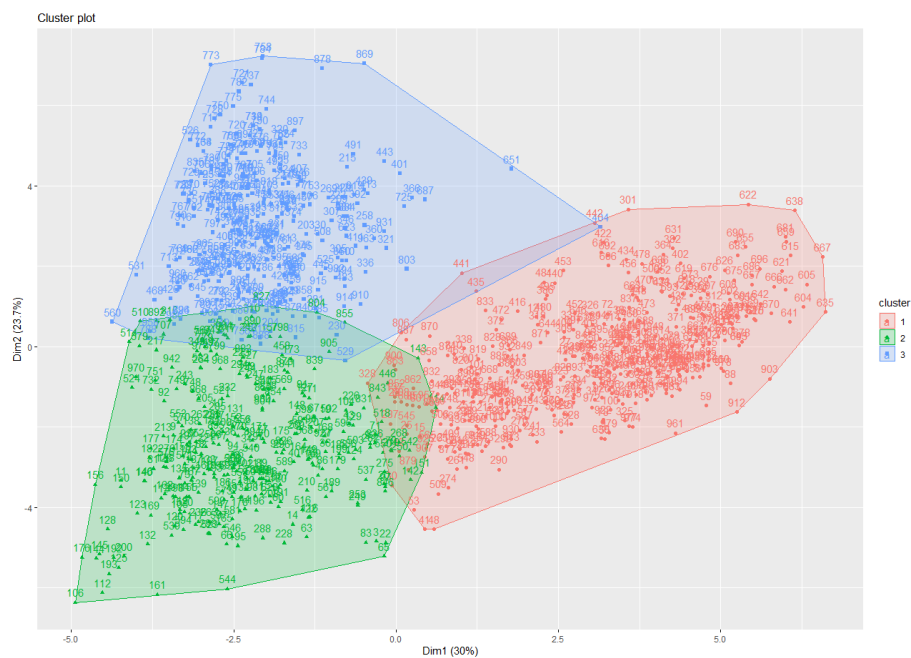


Figura 2.3: Clustering con metodo PAM

	1	2	3
blues	56	44	0
classical	5	95	0
country	36	43	21
disco	54	5	41
hiphop	43	3	54
jazz	23	56	21
metal	97	0	3
pop	0	9	91
reggae	38	21	41
rock	62	11	27

Figura 2.4: Tabella di confronto con k-means con le etichette a priori

I cluster ottenuti dopo aver applicato la PAM mostrano risultati del tutto coerenti con i precedenti, considerando che la numerazione dei cluster sia totalmente randomica. Utilizzando la funzione summary, notiamo come il valore massimo della silhouette raggiunge 0.25 relativamente ai tre cluster. Anche in questo caso concludiamo che non raggiunge il risultato desiderato e quindi conferma l'analisi precedente. A questo punto proviamo a utilizzare un metodo robusto, la trimmed k-means, per capire se la presenza di outlier abbia in qualche modo influenzato le analisi precedenti. Attraverso la funzione ctlcurves (di cui riportiamo il grafico sotto) si è visto che il numero di cluster ottimale rimane tre e che il trimming si aggira intorno al 3%.

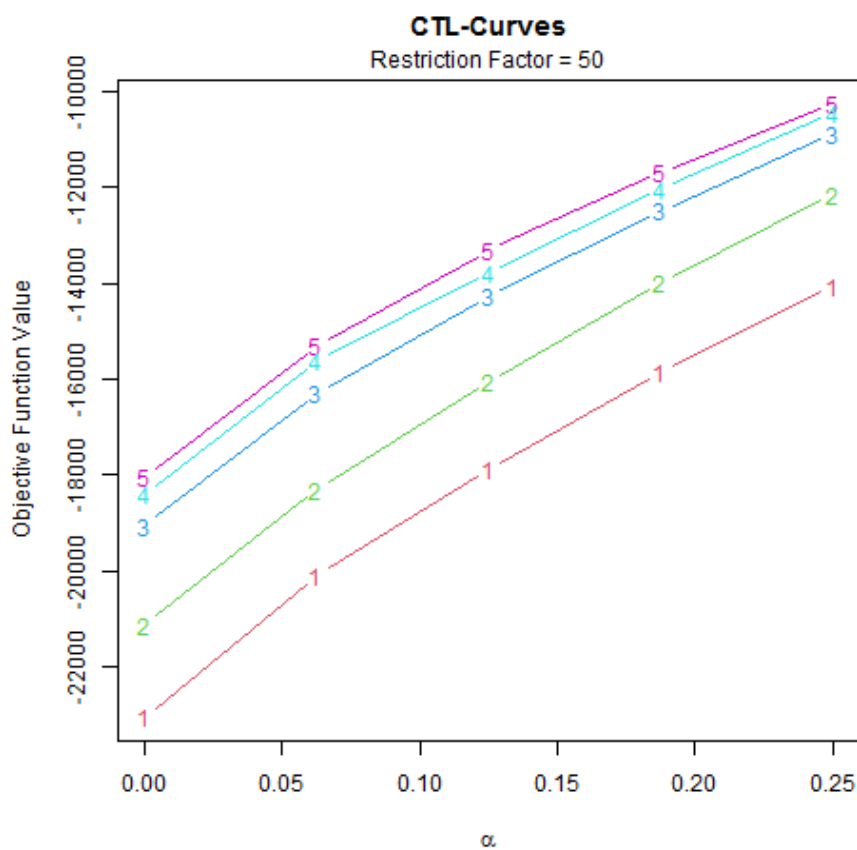


Figura 2.5: Ctlcurves

Con i dati appena ottenuti, applichiamo la trimmed k-means. I cluster ricavati sono i seguenti:

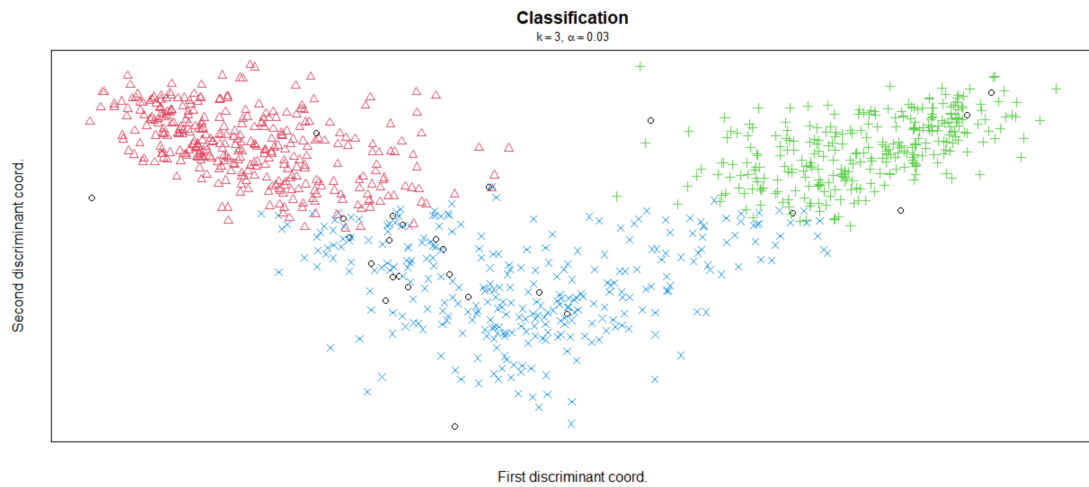


Figura 2.6: Grafico Trimmed K-Means

	0	1	2	3
<i>blues</i>	10	47	0	43
<i>classical</i>	8	1	1	90
<i>country</i>	0	25	24	51
<i>disco</i>	0	53	45	2
<i>hiphop</i>	1	41	53	5
<i>jazz</i>	0	13	23	64
<i>metal</i>	0	96	3	1
<i>pop</i>	4	0	91	5
<i>reggae</i>	2	25	44	29
<i>rock</i>	0	52	29	19

Figura 2.7: Tabella di confronto con trimmed k-means con le etichette a priori.

Il cluster 0 corrisponde ai dati trimmati.

Sono state calcolate le distanze euclidee dai centroidi dei valori etichettati come outlier. Abbiamo voluto verificare quali valori avessero una distanza maggiore del percentile 0.975 della χ^2 : i valori che non rispettano questa condizione (con distanza inferiore) risultano *swamped*. Il risultato è stato che solo tre valori su venticinque del trimming sono stati erroneamente etichettati.

3. Principal Component Analysis (PCA)

In questo capitolo abbiamo analizzato un approccio di riduzione dimensionale attraverso l'analisi delle componenti principali.

Abbiamo scoperto che sei componenti principale determinano l'80% della varianza totale, pertanto decidiamo di effettuare la k-means (robusta e non) con queste sole componenti.

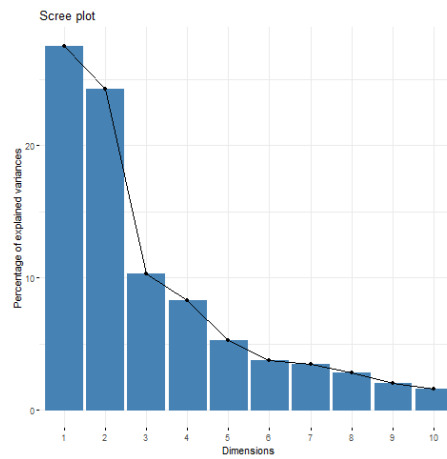


Figura 3.1: Contributi sulla varianza totale delle componenti principali

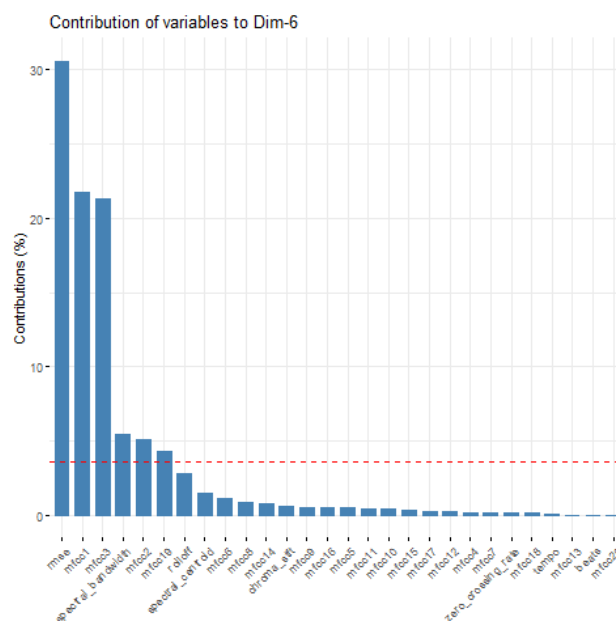


Figura 3.2: Contributo delle nostre feature a sei componenti principali

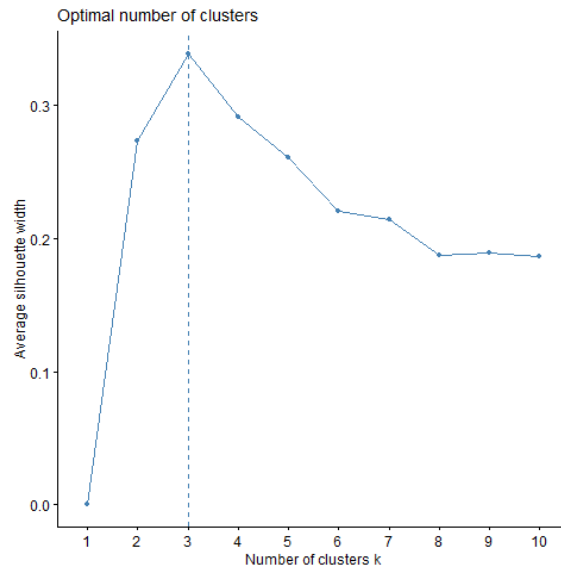


Figura 3.3: Silhouette K-Means con sei componenti principali

Applicando la funzione k-means con sei componenti principali ci accorgiamo che i valori della silhouette migliorano ma non abbastanza da giustificare una struttura di gruppo forte.

Applichiamo lo stesso ragionamento anche per la trimmed k-means:

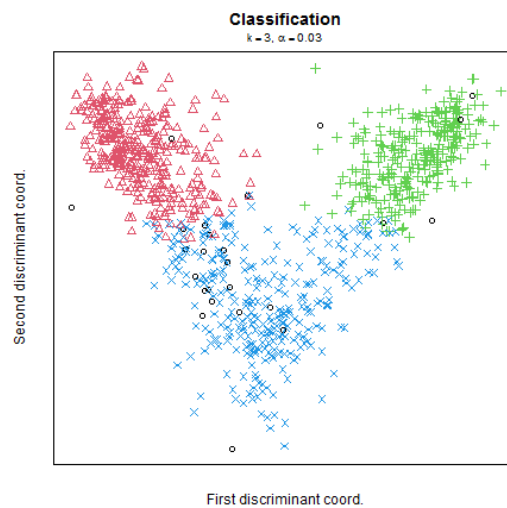


Figura 3.4: Clustering della trimmed k-means con sei componenti

	0	1	2	3
<i>blues</i>	10	47	0	43
<i>classical</i>	8	1	1	90
<i>country</i>	0	25	24	51
<i>disco</i>	0	53	45	2
<i>hiphop</i>	1	41	53	5
<i>jazz</i>	0	13	23	64
<i>metal</i>	0	96	3	1
<i>pop</i>	4	0	91	5
<i>reggae</i>	2	25	44	29
<i>rock</i>	0	52	29	19

Figura 3.5: Tabella di confronto della trimmed k-means con sei componenti

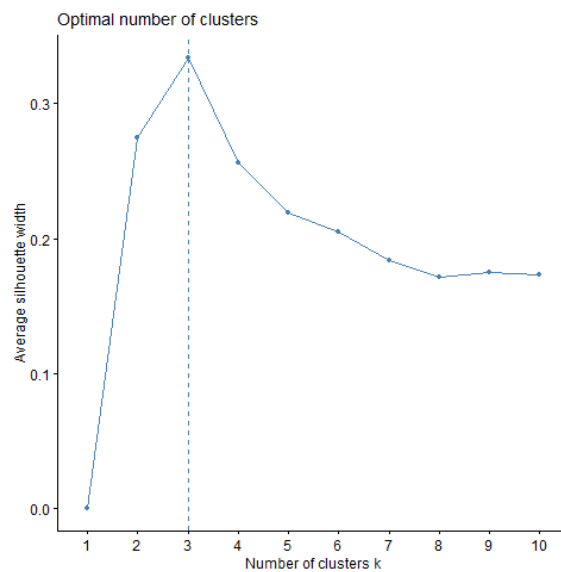


Figura 3.6: Silhouette della Trimmed K-Means con sei componenti principali

Conclusioni

Possiamo concludere che le feature presenti nel dataframe non permettono di effettuare una clusterizzazione ottimale automatica per generi musicali basandoci solamente sullo spettro di frequenze e di potenze.

Ad eccezione di tre generi (pop, classical e metal) i rimanenti sono distribuiti tra tutti i cluster (vedi reggae).

Un'ipotesi sul perché quei generi sono ben classificati come si vede dal confronto con le etichette a priori, potrebbe trovarsi nel fatto che quei generi hanno delle sonorità estreme e facilmente riconoscibili anche da non esperti musicisti.

Infatti una rapida analisi effettuata su un sottoinsieme con solo quei tre generi restituisce una segmentazione più marcata, ma il valore di silhouette non migliora di molto.

La spiegazione per cui i cluster rimangono molto vicini o con diversi bridge point potrebbe trovarsi nel fatto che medesimi strumenti musicali vengono utilizzati in diversi generi; si noti infatti che i tre generi che si distinguono molto sono quelli che presentano delle strumentazioni musicali più caratteristiche.

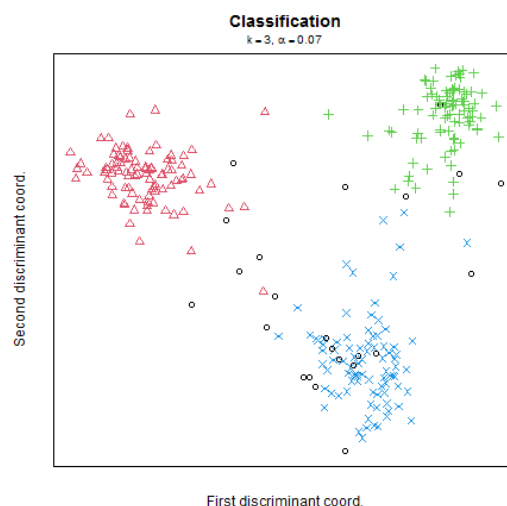


Figura 3.7: Clustering con Trimmed K-Means del sottoinsieme dei soli tre generi pop, metal e classical