



Università degli Studi di Milano - Bicocca
Machine Learning
Progetto 2022

Machine learning project

Progetto ML: Multiclass music genre classification

Studenti: Alessandro Albi, Daniel Scalena - 817769, 844608

Corso: Machine Learning - d.scalena@campus.unimib.it & a.albi1@campus.unimib.it

Contents

1 Overview dei dati	3
2 Exploratory data analysis	4
2.1 Dimensioni del dataset	4
2.2 Valori mancanti e non numerici	4
2.3 Analisi per attributo	5
2.4 Analisi bivariata	18
3 PCA	20
3.1 Correlazione tra features	20
3.2 Risultati e considerazioni sui principal component	22
3.3 Split del dataset	25
4 Support Vector Machine	26
4.1 Ricerca dei migliori parametri	26
4.2 Risultati ottenuti	26
5 Neural Network	28
5.1 Specifiche e problematiche della rete neurale	28
5.2 Risultati ottenuti	28

6 Feature engineering and second Neural Network	30
6.1 Bag of Words sul titolo delle tracce	30
6.2 Idee per l'impiego e sviluppo di una nuova rete neurale	32
6.3 Nuovi risultati ottenuti	32
7 Final considerations	34
7.1 Considerazioni temporali	34
7.2 Confronto tra metriche	34

1 Overview dei dati

Lo scopo del progetto è la costruzione di un sistema di classificazione in grado di distinguere 10 diversi generi musicali. I dati, principalmente numerici, riescono a raccogliere una serie di informazioni relative ad ogni traccia. Date queste caratteristiche è possibile addestrare diversi tipi di modelli supervisionati vista la presenza della classe di appartenenza per ogni traccia musicale del dataset.

Il progetto può essere diviso in più fasi: come primo passaggio viene esplorato il dataset per comprenderne le caratteristiche di ogni attributo. Questo passaggio è svolto osservando i valori numerici, costruendo grafici a riguardo per ogni tipo di osservazione che si vuole effettuare ed, eventualmente, effettuando statistiche appropriate al genere di dato che si sta osservando. Successivamente a questa fase si passa alla PCA in grado di fornire informazioni utili riguardo la diversa distribuzione di varianza del dataset. Ci si chiede quindi se può essere utile o meno ridurre le dimensioni effettive del dataset riassumendo più attributi in *principal components* che descrivono allo stesso modo la varianza ma riducendo contemporaneamente la quantità di attributi.

Successivamente ad una fase puramente esplorativa si passa al vero e proprio addestramento dei diversi modelli scelti. Nel nostro caso, vista la composizione del dataset e il numero totale di istanze disponibili, si è scelto di eseguire l'addestramento di una macchina a vettori di supporto e di una rete neurale per cercare di predirre al meglio le classi rappresentate i generi musicali. Per ogni modello impiegato ne saranno descritte le caratteristiche, la metodologia di impiego e successivamente discussi i vari risultati ottenuti.

Proprio in merito ai risultati viene effettuato un ulteriore lavoro di feature engineering vista la presenza di scarse performance sul dataset. Vengono quindi riconsiderati e successivamente analizzati gli attributi esclusi in un prima fase di esplorazione del dataset. Ogni dettaglio relativo a questa fase è esaustivamente spiegato nel relativo capitolo.

Vengono infine effettuate delle considerazioni finali riguardo la classificazione effettuata e i modelli che risultano spiccare in termini di performance. Vengono anche analizzate ulteriori informazioni riguardo il tempo di impiego e infine forniti importanti dettagli su come la classificazione stessa possa migliorare con strumenti non a nostra disposizione per il progetto stesso.

2 Exploratory data analysis

È stata eseguita un'analisi esplorativa dei dati e, contemporaneamente, un preprocessing degli stessi per pulire ogni istanza da eventuali problematiche legate ad esempio alla presenza di valori nulli. Di seguito per ogni attributo ne vengono elencate le caratteristiche e i vari test effettuati:

2.1 Dimensioni del dataset

Il dataset in nostro possesso ha un numero di attributi pari a 18 e un totale di 50 005 istanze. È possibile notare quindi una grande presenza di elementi che permetteranno sicuramente di usare diverse tecniche di apprendimento disposte ad accettare una grande quantità di dati.

Sono stati ulteriormente scartati diversi attributi per l'apprendimento relativi a un indice di traccia, del tutto ininfluente per il task di classificazione, e ulteriori due attributi riguardanti la data di ottenimento della specifica canzone, anche questa poco esplicativa sul genere di musica della stessa. Si passa quindi da 18 a 16 attributi utili.

2.2 Valori mancanti e non numerici

Eseguendo i dovuti calcoli è possibile notare come siano presenti dei valori nulli per alcuni attributi nel dataset. La loro numerosità è pari a 5 che, se confrontata con il ben più alto numero di istanze, risulta essere poco significativa. Si è pertanto deciso di eliminare queste istanze portando quindi il numero totale di elementi a 50 000.

Nel corso dell'esplorazione riassuntiva fornita dalla funzione *summary* di R è stato possibile identificare valori ambigui per quanto riguarda l'attributo *tempo*. Nello specifico, lo stesso dovrebbe contenere valori esclusivamente numerici in virgola mobile ma in alcuni casi è presente un simbolo ? che denota chiaramente la mancanza dello specifico valore. La numerosità di questi valori, rispetto alle dimensioni del dataset, è risultata essere pari a circa il 10%, pari quindi a circa 5 000 diverse istanze. Vista la volontà di mantenere le importanti informazioni a riguardo si è deciso di sostituire questi valori mancanti con la mediana della stessa colonna. Si è scelta questa misura, a fronte della media, per evitare di includere nella statistica eventuali outliers.

2.3 Analisi per attributo

Successivamente ad una prima analisi ed esplorazione globale del dataset si è cercato di scendere nel dettaglio per analizzare al meglio ogni attributo del dataset.

Il primo attributo analizzato è stato quello relativo all'etichetta di ogni canzone. Ci troviamo davanti ad un dataset bilanciato, vista la presenza di 5 000 istanze per ogni attributo. In totale sono presenti 10 attributi diversi: '*Electronic*', '*Anime*', '*Jazz*', '*Alternative*', '*Country*', '*Rap*', '*Blues*', '*Rock*', '*Classical*', '*Hip-Hop*'.

Di seguito vengono forniti nel dettaglio i vari passaggi effettuati per gli attributi dedicati all'addestramento dei modelli:

- **track_name**: il primo attributo analizzato contiene il nome della traccia. È ovviamente interpretato come una stringa (class *character*). Ad un primo sguardo si è pensato di eliminare questo attributo vista la difficile interpretazione da parte dei modelli di tale classe. Successivamente però si è notato come la lunghezza di questo attributo potesse in realtà fornire informazioni utili alla classificazione. Si è quindi creato un nuovo attributo, chiamato *length_title* che contenesse il numero di caratteri presenti nella titolo della traccia. Osservando il boxplot prodotto in figura 1 è possibile osservare come questa tecnica riesce a distinguere la classe relativa alla musica classica rispetto alle altre classi. Effettivamente, andando ad analizzare i titoli delle tracce classiche, è possibile osservare come siano mediamente più lunghe rispetto a tutte le altre canzoni nel dataset.

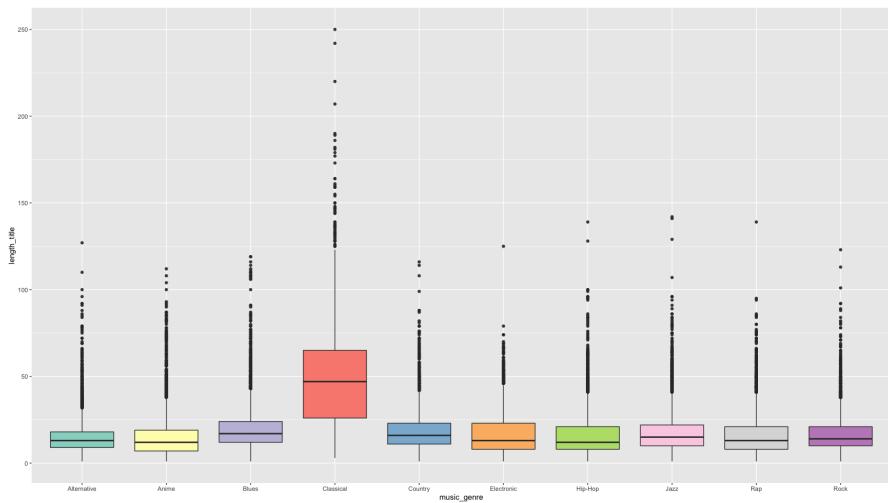


Figure 1: len title boxplot

La differenza descritta può essere notata anche nell'istogramma in figura 2.

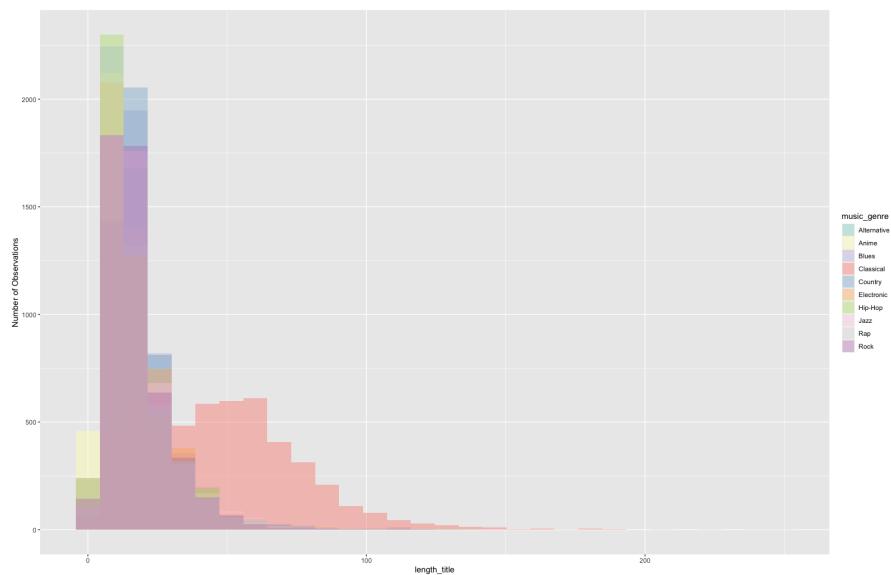


Figure 2: len title histogram

Questa colonna sarà successivamente frutto di altre tecniche di Feature engineering, adottando tecniche elementari di Natural Language Processing, per migliorare la classificazione. Tutti i dettagli sono descritti nel capitolo 6.

- **popularity:** questo attributo descrive la popolarità di una specifica traccia. È possibile osservare come sia un valore continuo compreso nell'intervallo [0, 100]. Vista la natura dell'attributo viene prodotto un boxplot in figura [3] dove è possibile osservare le diverse distribuzioni divise per ogni classe riguardante il genere musicale. Anche qui notiamo una certa varianza tra classi che potrà essere ben spiegata e interpretata dai modelli in fase di addestramento.

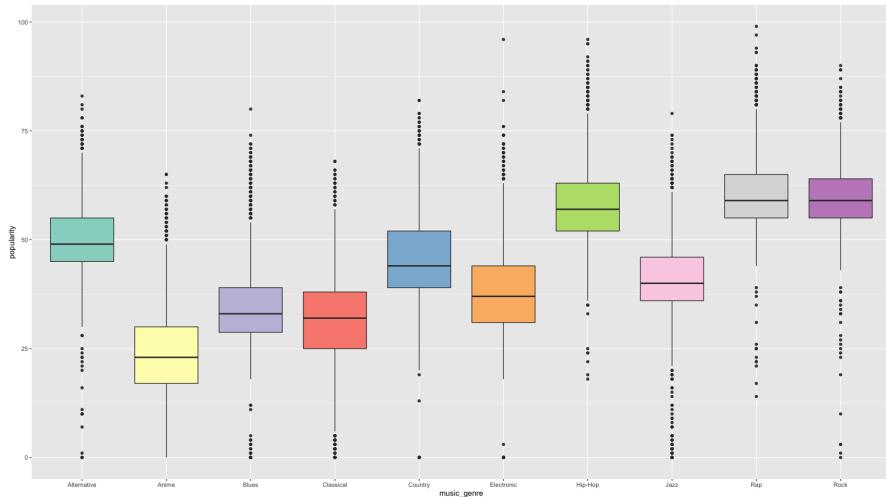


Figure 3: popularity boxplot

- **acousticness:** questo attributo è compreso in un intervallo [0, 1] e descrive l'acustica di un determinata traccia. Rispettando la complessità dei generi musicali si può notare come questo valore cerca di riassumere le frequenze e i picchi utilizzati durante la registrazione di un traccia. Anche se non perfettamente corretto può essere interpretato come un valore che descrive quanto una traccia sia parlata/cantata rispetto ad una traccia esclusivamente composta da musica senza la presenza di voci.

Dal boxplot prodotto in figura [4] è possibile osservare come per la musica classica e per il jazz il valore di mediana risulta essere decisamente più alto rispetto agli altri generi musicali. Questo avviene perchè, come

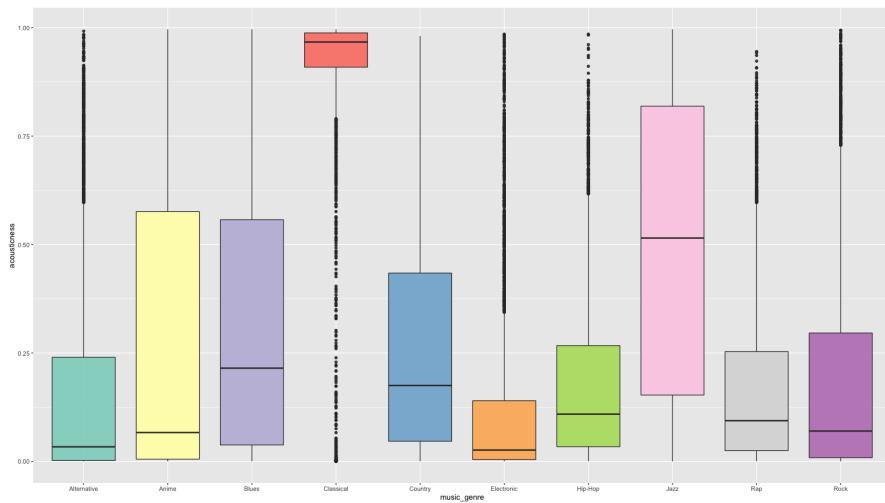


Figure 4: acousticness boxplot

facilmente intuibile, in questi generi musicali la presenza di voci è decisamente limitata rispetto ad esempio all'hip-hop, al rap e al rock. Ci si aspetta quindi che questo attributo aiuti le performance della musica classica e della musica jazz rispetto agli altri generi. Evidenze di questo tipo sono ulteriormente visibili nell'istogramma prodotto in figura 5.

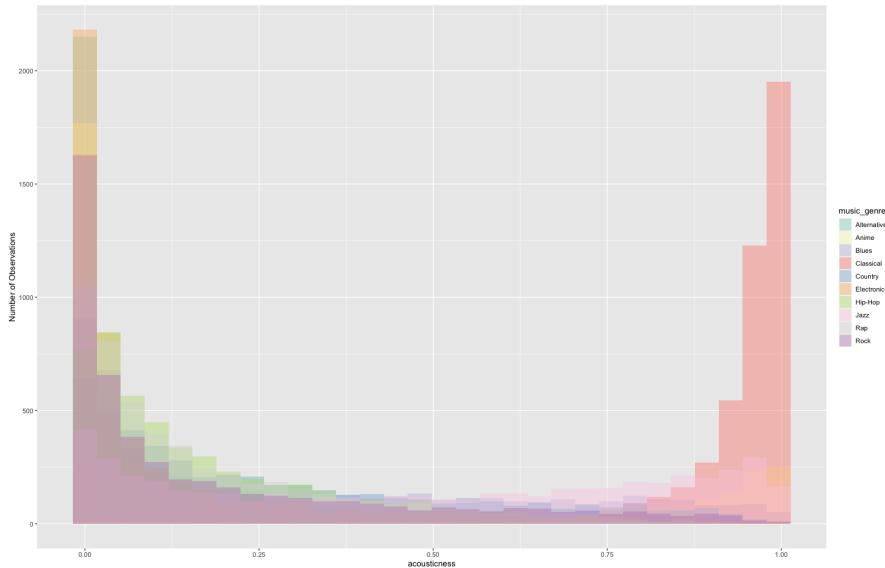


Figure 5: acousticness histogram

- **danceability**: per questo attributo osserviamo un valore numerico, sempre compreso tra 0 e 1, che descrive quanto una determinata traccia si propensa alla danza o al ballo. Ignorando la provenienza di questi dati è difficile comprendere come questo attributo sia stato determinato ma osservando il boxplot in figura [6] notiamo come anche questo riesca abbastanza bene a distinguere la musica classica da tutte le altre classi vista la presenza di un intervallo decisamente minore rispetto al restante dei generi. Discorso diverso invece può essere fatto per le altre tipologie di musica che rimangono difficilmente distinguibili con gli attributi fin'ora presi in analisi.

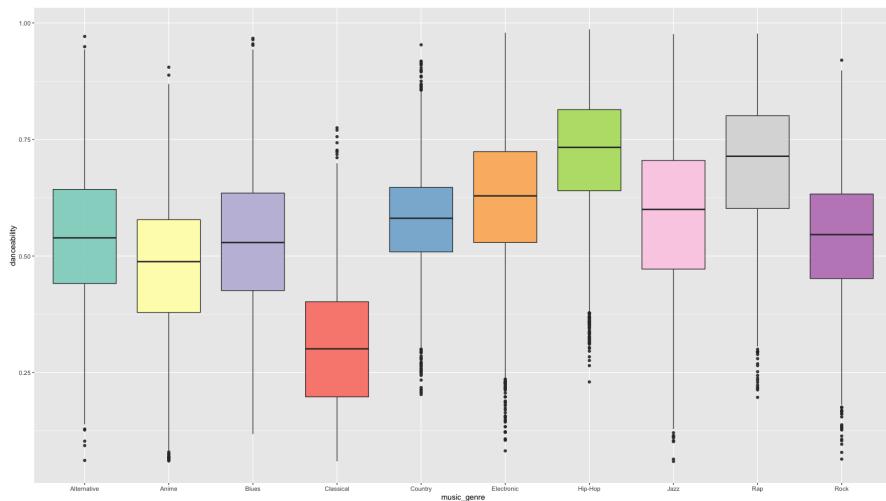


Figure 6: danceability boxplot

- **duration_ms**: come facilmente intuibile questo attributo descrive la durata di una traccia espressa in millisecondi per una maggiore precisione.

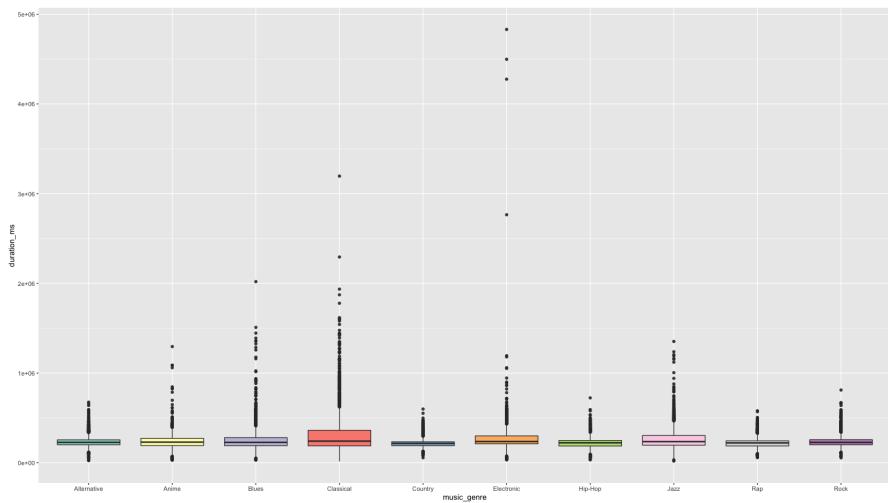


Figure 7: duration boxplot w/ outliers

Il boxplot prodotto in figura [7] risulta essere poco esplicativo vista l'elevata presenza di outliers. Si è quindi deciso di rimuovere temporaneamente gli outliers ottenendo un boxplot come in figura [8].

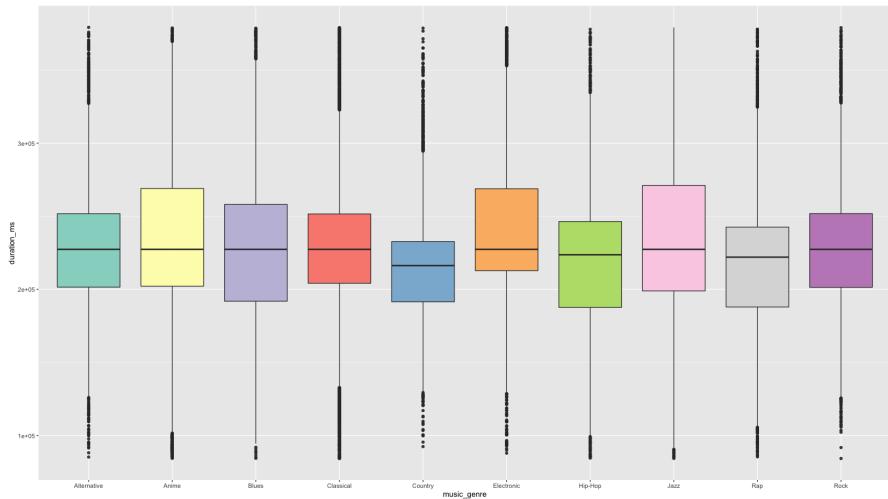


Figure 8: duration boxplot w/out outliers

Purtroppo, come evidente, non è presente una grande differenza tra classi, rendendo questo attributo decisamente poco esplicativo nei confronti del genere musicale e quindi poco utile per un task di classificazione.

- **energy**: questo attributo descrive l'energia di un traccia e come questa sia trasmessa all'ascoltatore. Si ignora purtroppo la provenienza di questa statistica ma, vista la natura soggettiva, è possibile pensare come sia frutto di un sondaggio svolto facendo ascoltare le diverse canzoni.

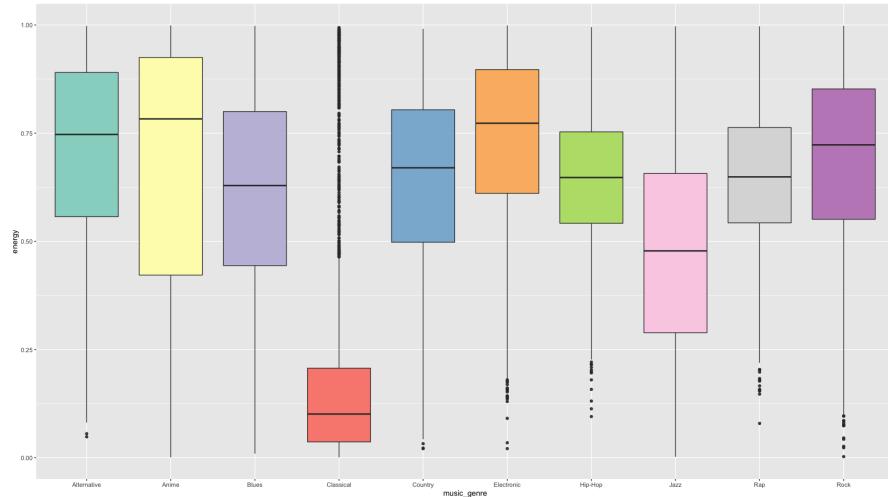


Figure 9: energy boxplot

Indipendentemente dalla provenienza dell'attributo è osservabile in figura [9] come l'energia sia espressa con un valore numerico compreso nell'intervallo $[0, 1]$ e come, anche questa volta, solo la musica classica riesce di molto a distinguersi dagli altri generi musicali. Notiamo tuttavia una varianza sicuramente maggiore rispetto al precedente attributo tra le varie classi.

- **instrumentalness**: questo attributo, a differenza degli altri visti precedentemente, risulta essere decisamente anomalo come osservabile in figura [10]. Osservando le varie statistiche è possibile notare come l'intero primo quartile di tutte le osservazioni sia pari a 0.

Vengono quindi calcolate il numero di stanze che hanno questo attributo pari a 0 e si osserva come il 30% delle informazioni hanno il valore nullo. Vista la natura sconosciuta dell'attributo e la grande mancanza di informazioni fornite per un task di classificazione si è deciso di scartare

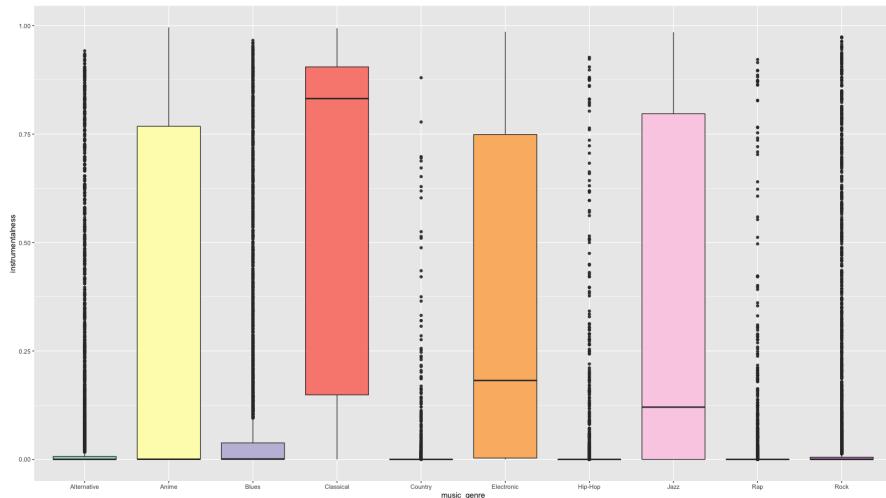


Figure 10: instrumentalness boxplot

l'intera colonna riducendo ancora di una volta il numero di informazioni utili all'addestramento.

- **Key:** Per questo attributo categorico è stato verificato come errato per la maggior parte di canzoni come incluse nel dataset. A quanto visto si tratta semplicemente di una stima effettuata con tool automatici senza considerare che molto spesso le chiavi cambiano durante lo scorrere della canzone. Non viene pertanto considerato ai fini della classificazione
- **Liveness:** questo attributo cerca di descrivere la vitalità della traccia con un attributo anche in questo caso che varia tra 0 e 1, probabilmente anche questa volta frutto di un sondaggio vista la natura soggettiva dello stesso. Come osservabile dal boxplot in figura [11] ci troviamo ancora una volta davanti ad un attributo che riesce a descrivere poco le differenze tra i vari generi musicali. Si decide tuttavia di conservare tale attributo per sfruttarne trasformazioni operate durante la fase di addestramento dei vari modelli.

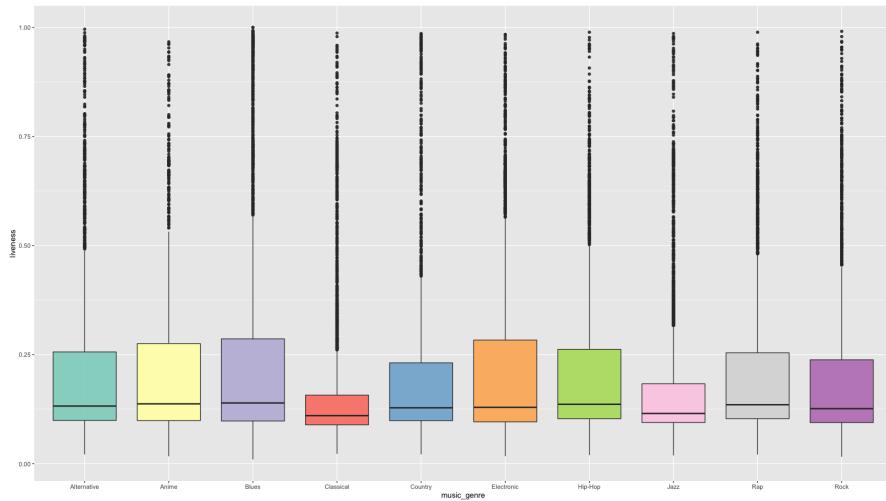


Figure 11: liveness boxplot

- **Loudness:** questo attributo deriva da una misurazione media in decibel riguardo alla registrazione della traccia. Per costruzione uno studio di registrazione decide come effettuare il mastering di una canzone modulando l'intensità e gestendo la coordinazione dei vari strumenti e voci coinvolte. Dai valori numerici osservabili nel boxplot in figura [12], l'unica classe che riesce a distinguersi è ovviamente quella relativa alla musica classica che difatti viene generalmente registrata con un'intensità più bassa rispetto ad altre canzoni destinate ad un uso più popolare. Lievi differenze sono inoltre osservate per la musica jazz e blues.

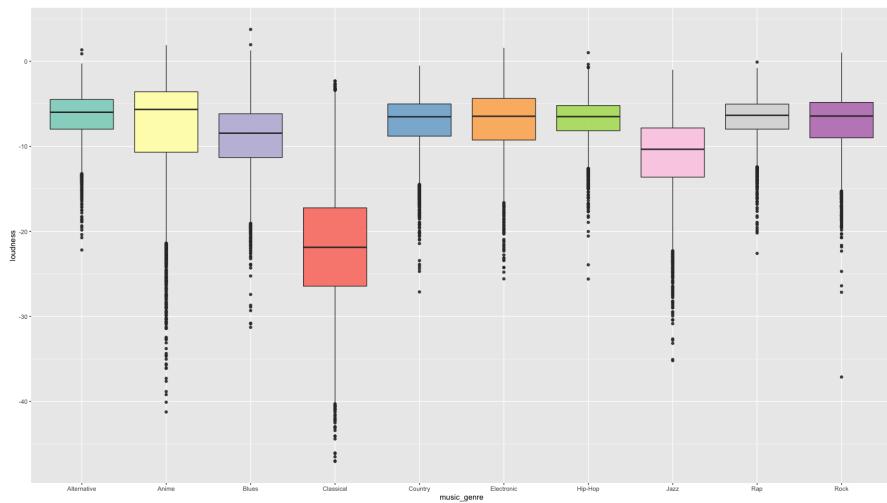


Figure 12: loudness boxplot

Evidenze a riguardo sono osservabili anche nella distribuzione in figura 13.

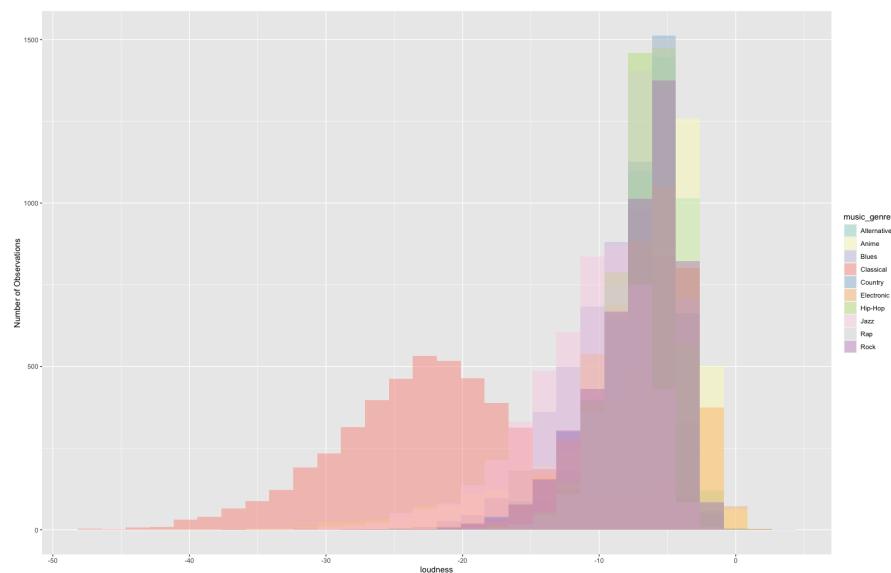


Figure 13: loudness histogram

- **Mode:** [TODO]
- **Speechiness:** come per *acousticness* questo attributo misura quanto una traccia si parlata/cantata rispetto ad una prevalentemente composta esclusivamente da musica. Nel boxplot prodotto in figura [14] spiccano sicuramente le classi relative ai generi Hip-Hop e Rap, con valori decisamente più alti rispetto a tutti gli altri generi musicali.

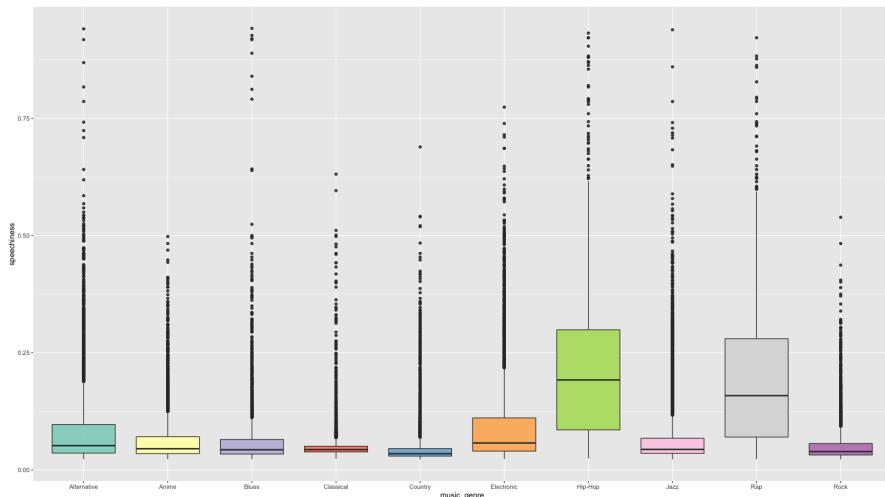


Figure 14: speechiness boxplot w/ outliers

Vista la grande presenza di outliers, l'ulteriore boxplot in figura [15] descrive lo stesso attributo evidenziandone le differenze.

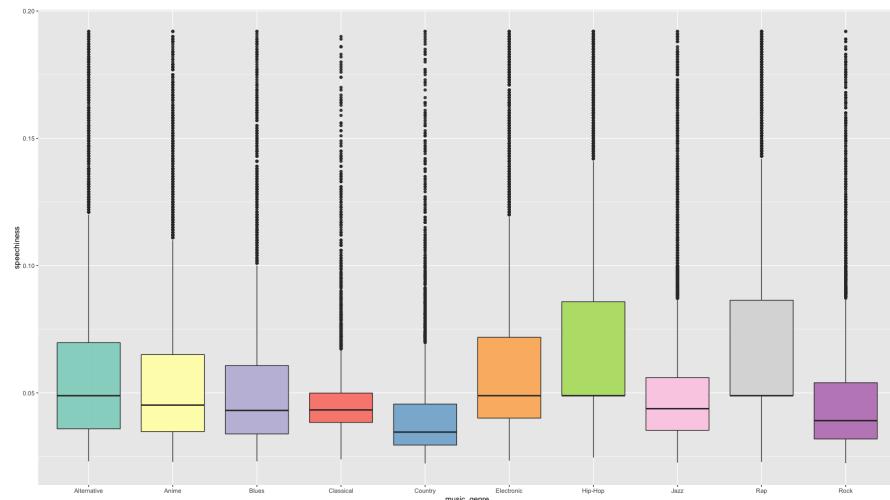


Figure 15: speechiness boxplot w/out outliers

- **Tempo**: come facilmente intuibile questo attributo usa la nozione di tempo musicale per identificare le tracce. Come già visto precedentemente, anche in questo caso come osservabile dal boxplot in figura [16] le differenze riguardano principalmente la musica classica. In questo caso i dati corrispondono sicuramente alla realtà visto che quasi tutte le moderne canzoni sono registrate con un tempo nell'intorno dei 120bpm (confermato dalla media e dalla mediana entrambe con un valore di circa 119).

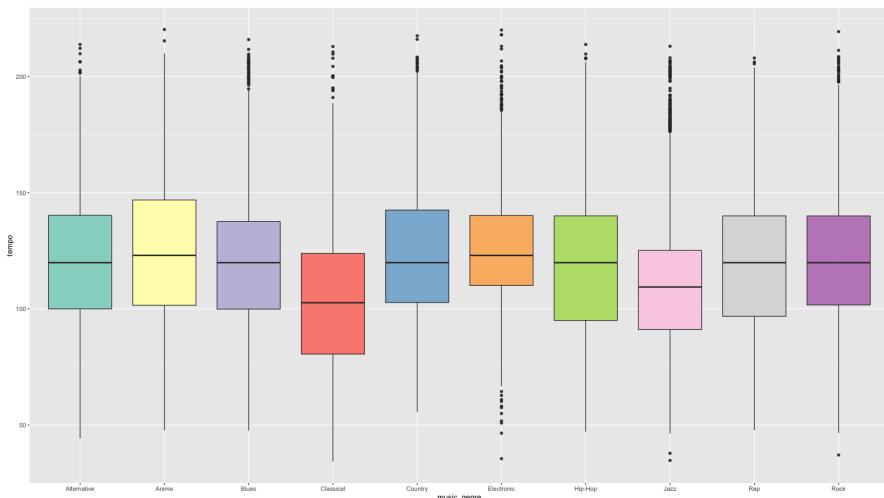


Figure 16: tempo boxplot

- **valence**: ultimo attributo analizzato riguarda la valenza di una traccia. In questo caso è un valore numerico tra 0 e 1 che descrive quanto una traccia crea sensazioni positive (valore più alto) rispetto a quelle negative (valore più basso).

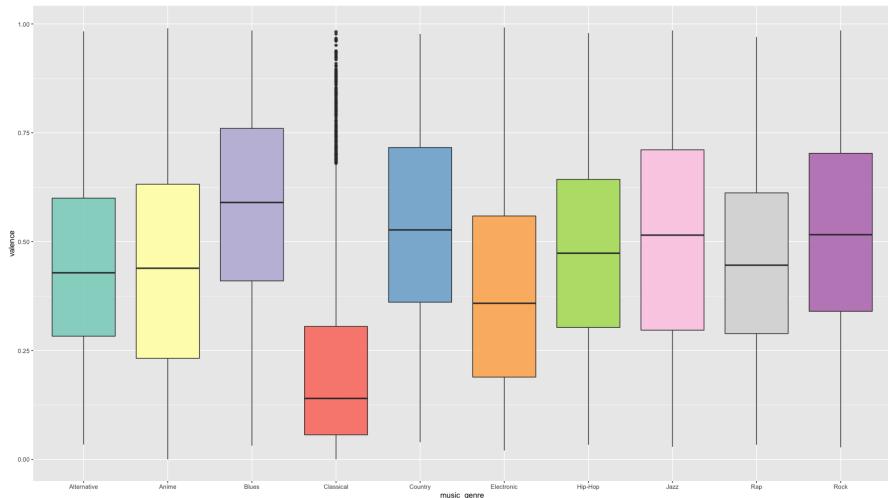


Figure 17: valence boxplot

Come osservabile dal boxplot in figura [17] anche questa volta la musica classica è l'unica che riesce davvero a distinguersi, essendo generalmente più triste e calma. I valori più alti invece sono per i generi Blues, Country, Jazz e Rock.

2.4 Analisi bivariata

È possibile costruire degli scatterplot in grado di identificare come due variabili siano reciprocamente distribuite in un asse cartesiano. Gli scatterplots in figura 18 evidenziano come, quanto affermato precedentemente nelle statistiche univariate, risulti essere vero anche in caso di analisi bivariata (Da sinistra verso destra e dall'alto verso il basso: *danceability-acousticness*, *energy-acousticness*, *loudness-energy*, *loudness-liveness*). L'unica classe a risultare ben distinguibile dalle altre è quella del genere di musica classica.

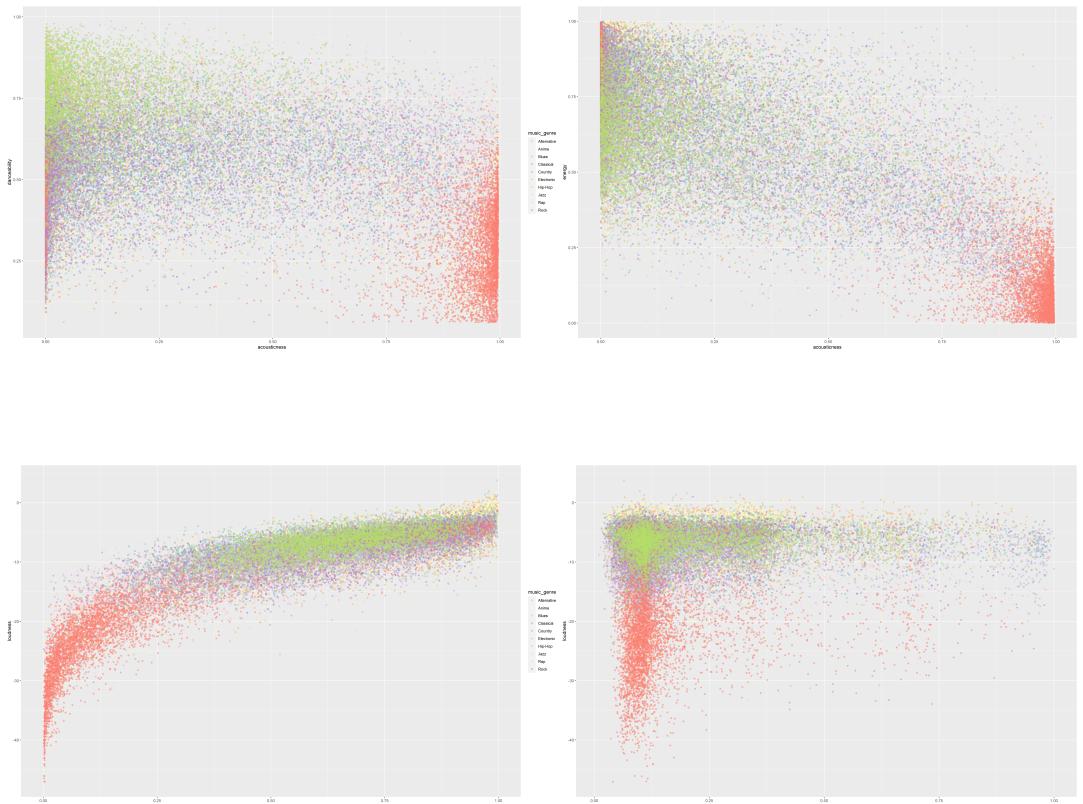


Figure 18: Scatter plot per diversi attributi. Il colore rosso è associato al genere di musica classica, tutti gli altri colori, meno visibili vista la quantità di classi, sono associati agli altri generi musicali da classificare.

Altri scatterplot invece confermano come i dati non diano informazioni significative su nessuna classe nello specifico, rendendo il task di classificazione sicuramente più complicato e sicuramente lontano da ottime performance.

Si noti comunque una possibile correlazione tra energy e loudness (tra i grafici presentati in basso a sinistra) o una mancanza di correlazione per gli altri attributi.

Non si esclude tuttavia una maggiore comprensione dei dati da parte degli algoritmi di classificazione che verranno introdotti, capaci di combinarli per trovare corrispondenze a prima vista nascoste con tutti gli altri generi musicali.

3 PCA

Considerato l'elevato numero di attributi del dataset, viene normale chiedesi se effettuare o meno una riduzione della dimensionalità per diminuire il costo computazionale dell'addestramento dei modelli.

A tale scopo viene sfruttata la PCA(Principal Component Analysis), tecnica per la semplificazione dei dati che si basa sull'idea che all'interno dell'insieme dei dati siano presenti uno o più attributi ridondanti, ovvero che esprimono informazioni già fornite da altri attributi. La PCA utilizza una trasformazione ortogonale per convertire le osservazioni in un insieme di variabili linearmente non correlate, definite componenti principali. Pertanto si analizzerà quanto sia conveniente o meno andare a diminuire la numerosità degli attributi, senza perdere informazioni dal dataset.

L'analisi viene eseguita sul dataset escludendo gli attributi non numerici, pertanto si avrà un insieme avente 11 attributi.

3.1 Correlazione tra features

Per prima cosa bisogna studiare ed analizzare la correlazione tra i vari attributi.

Viene generata la matrice (figura [19]) che mostra la correlazione tra le varie features tramite variazioni cromatiche come riportate in agenda. Inoltre, all'interno di ogni incrocio viene riportato il valore di tale variazione

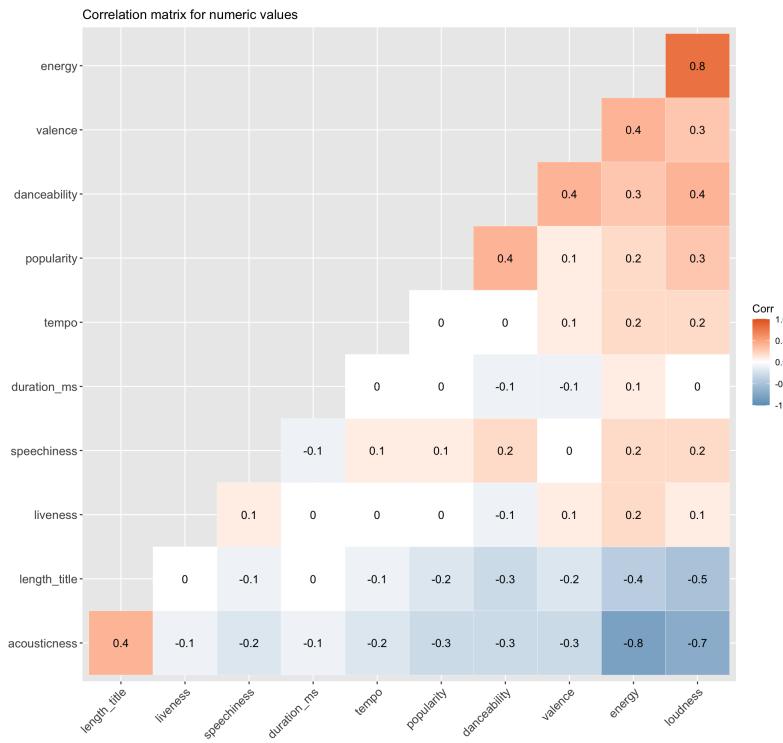


Figure 19: Correlation matrix

Dalla matrice si evince una forte **correlazione diretta** tra le features *energy* e *loudness*. Allo stesso modo si può notare una forte **correlazione indiretta** tra le features *acousticness* e *energy* e le feature *acousticness* e *loudness*.

A questo punto si applica la PCA al nostro insieme di dati, ottenendo la tabella riportata [1]. I risultati rappresentano l'autovalore di ogni dimensione, ovvero la quantità di varianza spiegata. Maggiore è l'autovalore di una dimensione, più essa è rilevante.

pred	eigenvalue	variance percent	cumulative variance percent
Dim.1	3.5578039	32.343672	32.34367
Dim.2	1.3023699	11.839726	44.18340
Dim.3	1.0785425	9.804932	53.98833
Dim.4	0.9808640	8.916945	62.90528
Dim.5	0.9535021	8.668201	71.57348
Dim.6	0.8472323	7.702112	79.27559
Dim.7	0.7706874	7.006249	86.28184
Dim.8	0.6548674	5.953340	92.23518
Dim.9	0.4734266	4.303878	96.53906
Dim.10	0.2657594	2.415995	98.95505
Dim.11	0.1149445	1.044950	100.00000

Table 1: Matrice degli autovalori e della varianza cumulata

Dalla tabella si può notare come solo le prime 3 dimensioni abbiano l'autovalore ad esso associato con valore maggiore di 1. Inoltre notiamo che per raggiungere una varianza maggiore del 90% sono necessarie 8 dimensioni.

3.2 Risultati e considerazioni sui principal component

A questo punto l'ultima cosa da fare è valutare se ridurre o meno la dimensionalità del dataset.

Per prima cosa si analizza il **biplot della PCA** (figura [20])

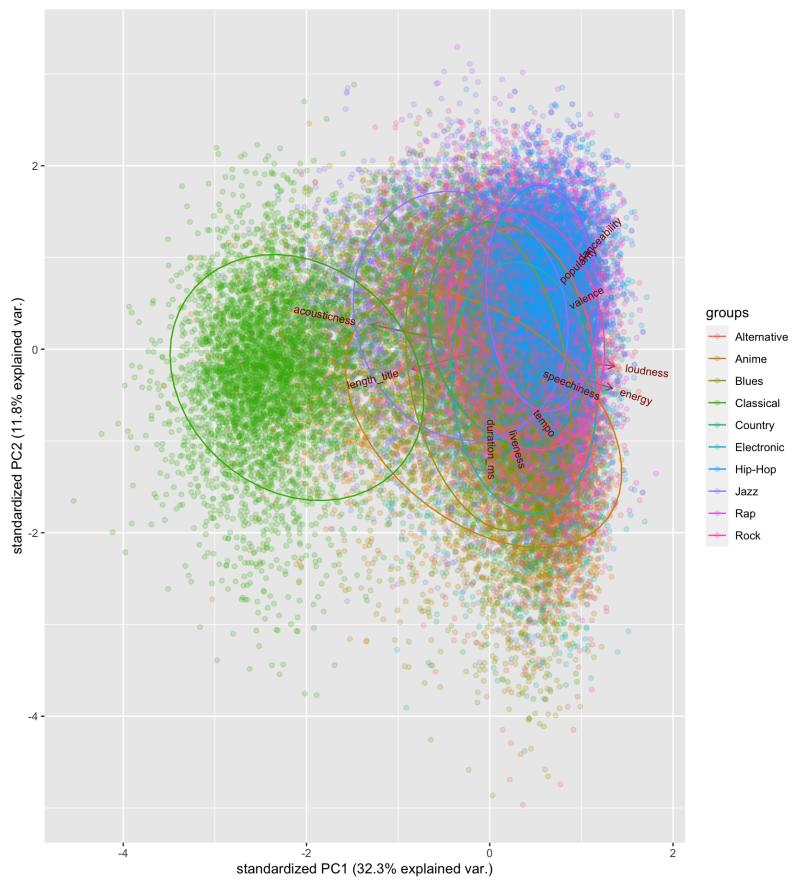


Figure 20: PCA biplot

Le due componenti principali (PC1 e PC2) esprimono il 45% circa della variazione totale dei dati. Da come si può notare in figura l'unica informazione utile che si può ottenere è la buona suddivisione della musica classica dagli altri generi musicali attraverso la PC1.

Come secondo step si analizza lo **Scree plot** in figura [21].

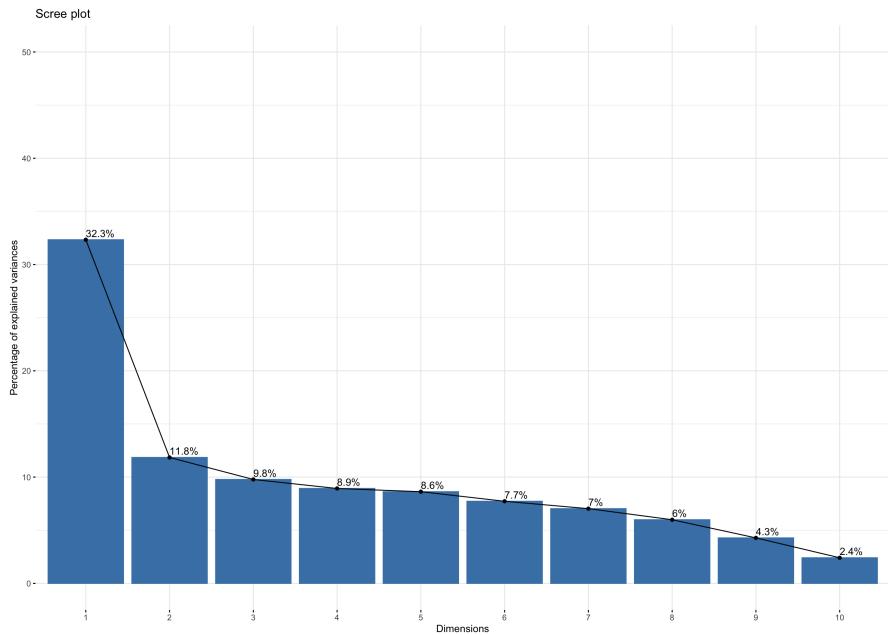


Figure 21: Scree plot

Questo è uno dei metodi utilizzati, insieme al controllo dei valori degli autovalori ed alla varianza cumulata, per capire quanto sia utile ridurre il numero delle feature e nel caso fosse utile, di quanto. Nel grafico si possono notare rappresentati gli autovalori in funzione del numero dei componenti principali. Essendo gli autovalori decrescenti, il grafico assume la forma di una spezzata discendente.

Da come si può notare, in questo caso, l'analisi dello Scree plot ci da' una chiara idea sulle componenti. In genere, in corrispondenza dell'ultima dimensione da considerare, dovrebbe esserci una brusca variazione di pendenza. Pertanto si può pensare che non sia una buona idea ridurre la dimensionalità del dataset, poiché la scelta ricadrebbe solo sulla prima componente, la quale esprime meno del 35% di varianza spiegata.

In conclusione si può affermare che, dopo un'attenta analisi delle features e delle loro correlazioni, bisognerebbe ridurre di poco la dimensionalità del dataset, di circa 3 features, per poter ottenere dei risultati accettabili. Ma questo risulta essere svantaggioso, perché sacrificare una piccola parte della dimensionalità perdendo una porzione utile di varianza porterebbe ad avere problemi in termini di informazioni ottenibili dal dataset.

3.3 Split del dataset

Come ultima fase viene effettuato uno split dell'intero dataset in un *train* e un *test* per permettere le successive fasi di addestramento dei modelli. Il dataset di *test* sarà esclusivamente usato in fase di test, pertanto le sue istanze non parteciperanno in alcun modo alla fase di addestramento e saranno completamente sconosciute al modello stesso. I criteri dello split sono di un 75% per il dataset di *train* e un restante 25% per il dataset di *test*, ottenendo rispettivamente 37 500 e 12 500 istanze per *train* e *test*.

Tutti i valori del dataset inoltre vengono automaticamente scalati prima di essere dati in input al modello. Viene pertanto usata una scala default con media pari a 0 e varianza unitaria.

4 Support Vector Machine

Il primo modello analizzato è stato una Support Vector Machine, in grado di separare le diverse classi in uno spazio lineare o non, in base al kernel utilizzato.

4.1 Ricerca dei migliori parametri

Viene effettuata una ricerca dei migliori parametri per il kernel da utilizzare (vengono scelti tra *lineare*, *polinomiale*, *radiale* e *sigmoid*). Questo processo è risultato essere decisamente dispendioso a livello di calcolo pertanto sono state ridotte momentaneamente le dimensioni del dataset di train (del 50%), assicurandosi comunque di mantenere lo stesso numero di classi anche nel dataset più piccolo.

Dai risultati ottenuti si è deciso di proseguire l'addestramento sull'intero dataset di train con un kernel di tipo *polinomiale* e un parametro di costo (C) pari a 1. Sono state ulteriormente mantenute le probabilità per le predizioni utili nella successiva fase riguardante l'analisi dei risultati.

4.2 Risultati ottenuti

Nella tabella 2 viene riportata la matrice di confusione sul dataset di test. Come è possibile osservare dalla stessa, l'accuratezza raggiunge un valore complessivo pari al 54%.

pred	Alternative	Anime	Blues	Classical	Country	Electronic	Hip-Hop	Jazz	Rap	Rock
Alternative	490	40	47	20	110	77	125	62	117	220
Anime	7	934	128	56	17	73	1	30	1	4
Blues	20	86	593	35	77	73	1	30	1	4
Classical	4	57	7	1037	1	7	0	42	0	3
Country	236	33	140	7	659	96	41	125	39	152
Electronic	89	74	97	30	56	737	19	141	14	8
Hip-Hop	92	1	4	0	24	50	599	30	515	28
Jazz	80	21	192	59	123	95	12	597	7	30
Rap	33	0	2	0	11	19	382	9	412	57
Rock	200	5	41	4	173	24	71	37	143	747

Table 2: Matrice di confusione per SVM dove la diagonale viene evidenziata

Ulteriori statistiche sono presenti in tabella 3 dove è evidente una buona classificazione solamente per alcune classi. In prima analisi, il genere musica

classica è quello meglio classificato riuscendo ad ottenere un punteggio F1 pari a 0.86 e un'area sotto la curva ROC pari a 0.88. Troviamo poi il genere di musica anime anch'esso con dei buoni risultati.

Come previsto nella prima fase esplorativa la musica classica riesce a ben distinguersi rispetto alle altre classi dove, i vari attributi, non si dimostrano efficaci nella distinzione dei vari generi musicali.

Stats	Balanced Accuracy	AUROC	precision	recall	F1
Alternative	0.6595	0.61	0.3916867	0.3746177	0.3829621
Anime	0.85921	0.96	0.7466027	0.7466027	0.7466027
Blues	0.71617	0.72	0.4740208	0.5583804	0.5127540
Classical	0.91009	0.88	0.8309295	0.8955095	0.8620116
Country	0.72477	0.87	0.5267786	0.4312827	0.4742713
Electronic	0.77110	0.43	0.5891287	0.5826087	0.5858506
Hip-Hop	0.70634	0.61	0.4788169	0.4460164	0.4618350
Jazz	0.71168	0.49	0.4783654	0.4909539	0.4845779
Rap	0.64227	0.41	0.3301282	0.4454054	0.3791993
Rock	0.76754	0.62	0.5971223	0.5169550	0.5541543

Table 3: Metriche per SVM dove sono evidenziati i risultati migliori ottenuti dal modello

Si conclude quindi che le macchine a vettori di supporto non riescono ad effettuare una classificazione precisa e puntuale dei vari generi musicali ma, consapevoli dei dati a disposizione, si può intuire come le basse performance siano probabilmente dovute alla scarsa qualità dei dati piuttosto che al modello scelto.

Nel capitolo 7 relativo alle considerazioni finali verranno comparati i modelli visti e le metriche ottenute con informazioni numeriche e grafici, determinando quale tra i modelli ha avuto delle performance migliori.

5 Neural Network

Il secondo modello scelto è una rete neurale in grado di apprendere le varie feature a disposizione per classificare i generi musicali. Prima di ogni addestramento e test effettuato, come anticipato nell'analisi dei risultati di SVM, ci si aspettano anche qui basse performance dovute alla scarsa qualità dei dati e alla loro poco espressività nelle differenze tra classi.

5.1 Specifiche e problematiche della rete neurale

La rete neurale addestrata, composta da due hidden layer di egual dimensioni pari a 10, riceve in input le 11 features numeriche del dataset e restituisce in output le 10 classi attraverso altri 10 neuroni. Il massimo numero di step effettuabili dalla rete è pari a $2e+05$ con un threshold pari a 2.2.

Dopo diversi test effettuati si è trovato il miglior compromesso con i parametri sopraelencati, considerando soprattutto i tempi di calcolo della stessa. Proprio quest'ultima problematica, vista anche la dimensione stessa dei dati, ha reso difficile e temporalmente oneroso l'addestramento della rete neurale, soprattutto per l'impossibilità in R di usare la computazione parallela con hardware dedicato come una GPU.

5.2 Risultati ottenuti

I risultati ottenuti sono stati salvati dopo una computazione pari a $1,2e+05$ steps con un tempo medio di calcolo di 9 sec ogni 100 step, per un totale di circa 3 ore.

Anche in questo caso sono state computate le metriche sul dataset di test, totalmente sconosciuto alla rete durante la fase di addestramento. La matrice di confusione in tabella 4 permette di capire come la rete si comporta per la classificazione dei diversi generi musicali. L'accuratezza generale raggiunta per la rete neurale è pari al 55.6%.

Ulteriori statistiche riguardo i risultati raggiunti dalla rete neurale sono presenti in tabella 5. Anche in questo caso la classe relativa al genere di musica classica risulta avere una buona performance rispetto a tutte le altre classi, proprio come ipotizzato nella prima fase esplorativa dei dati. Osservando i valori dell'area sotto la curva ROC è possibile notare come il modello compia un buon lavoro nel separare le varie classi pur commettendo errori di classificazione che

pred	Alternative	Anime	Blues	Classical	Country	Electronic	Hip-Hop	Jazz	Rap	Rock
Alternative	482	8	11	0	158	56	124	114	26	269
Anime	48	895	103	64	39	62	0	31	1	8
Blues	52	109	610	12	144	57	2	198	1	66
Classical	39	47	20	1061	8	29	0	38	0	7
Country	132	20	60	3	576	49	20	115	9	266
Electronic	122	72	87	6	85	671	32	111	20	45
Hip-Hop	35	1	0	0	13	14	852	21	236	79
Jazz	63	14	156	81	128	120	39	573	7	69
Rap	42	2	2	0	6	12	728	17	274	166
Rock	109	2	1	5	29	11	48	25	56	965

Table 4: Matrice di confusione per NN dove la diagonale viene evidenziata

portano a una non ottima accuratezza, precisione e recupero (ultime due riasumibili dal punteggio F1).

Stats	Balanced Accuracy	AUROC	precision	recall	F1
Alternative	0.68075	0.75	0.4288256	0.3862179	0.4064081
Anime	0.86677	0.94	0.7649573	0.7154277	0.7393639
Blues	0.76249	0.87	0.5809524	0.4876099	0.5302043
Classical	0.92226	0.98	0.8612013	0.8494796	0.8553003
Country	0.71305	0.89	0.4856661	0.4608000	0.4729064
Electronic	0.78497	0.79	0.6207216	0.5363709	0.5754717
Hip-Hop	0.71217	0.82	0.4617886	0.6810552	0.5503876
Jazz	0.70042	0.78	0.4609815	0.4584000	0.4596871
Rap	0.67639	0.61	0.4349206	0.2193755	0.2916445
Rock	0.73517	0.74	0.4974227	0.7713829	0.6048261

Table 5: Metriche per NN dove sono evidenziati i risultati migliori ottenuti dal modello

Anche in questo caso, come per i risultati visti su SVM, verranno mostrati in comparativa i vari modelli attraverso l'uso di grafici che ne chiariscono le differenze nel capitolo 7 riguardante le considerazioni finali.

6 Feature engineering and second Neural Network

I risultati ottenuti da SVM e rete neurale sono stati in grado di raggiungere gli stessi risultati ottenuti da altri utenti su Kaggle da dove il dataset deriva. Tuttavia risultano essere performance per niente buone rispetto a un task di classificazione. Viene quindi effettuato un lavoro a ritroso tornando ad analizzare le varie feature iniziali per capire come eventuali caratteristiche perse o nascoste possano migliorare la stessa classificazione.

Tra tutte le feature presenti è stata inizialmente poco analizzato l'attributo relativo al nome della traccia. Nel capitolo 2 viene creata una feature aggiuntiva che considera la lunghezza del titolo stesso creando quindi una variabile che, come visto, evidenzia le differenze della classe relativa alla musica classica rispetto agli altri generi musicali. Come visto però la musica classica è uno dei pochi generi ad essere ben classificato mentre tutti gli altri non riescono a raggiungere le stesse performance.

Se si analizzano le tracce con più dettaglio si può notare come alcune parole siano ricorrenti nei titoli, dipendentemente dal genere musicale. Proprio per questo motivo si è deciso di sfruttare quest'informazione aggiuntiva per permettere una migliore classificazione.

6.1 Bag of Words sul titolo delle tracce

Tra tutte le tecniche per il riconoscimento del linguaggio naturale probabilmente la bag of word (o BoW) risulta essere tra le più semplici ed elementari. Tuttavia nasconde potenzialità che possono dimostrarsi utile allo scopo di classificazione per il task preso in considerazione dal progetto.

Il loro funzionamento comprende la scansione del testo e la successiva analisi delle parole che sono più frequenti all'interno dello stesso. Si vuole quindi sfruttare questa potenzialità per estrarre tutte quelle parole che, dato un determinato genere musicale, risultano essere più utilizzate nei titoli delle tracce musicali in analisi.

Per ottenere delle bag of words il più pulite possibili tutte le stringhe, relativamente al nome delle tracce, vengono pulite da stopwords, whitespaces e tutti quei caratteri che risultano avere poco significato rispetto all'obiettivo che si vuole raggiungere.

Successivamente alla fase di pulizia vengono prese in considerazione tutte le parole che compaiono con una frequenza maggiore nei titoli. Come è pos-

sibile notare dalle wordclouds riprodotte in figura 22, 23, 24 e 25 sono presenti dei pattern per ogni genere musicale. Per fornire un esempio il genere musica elettronica presenta una grande quantità di termini *remix*, *edit* e così via nei loro titoli.

Si è successivamente deciso di mappare queste nuove informazioni in dummy variables aggiuntive che permettessero di specificare la presenza o meno di una determinata parola all'interno del titolo. Si ottiene quindi un nuovo dataset con, oltre alle 11 feature numeriche iniziali, anche 27 nuove variabili booleane in grado di segnalare quanto prima specificato. Viene aggiunta un'ulteriore variabile dummy in caso siano presenti simboli appartenenti all'alfabeto giapponese, caratteristica spesso ritrovata nei titoli delle tracce con il genere Anime.

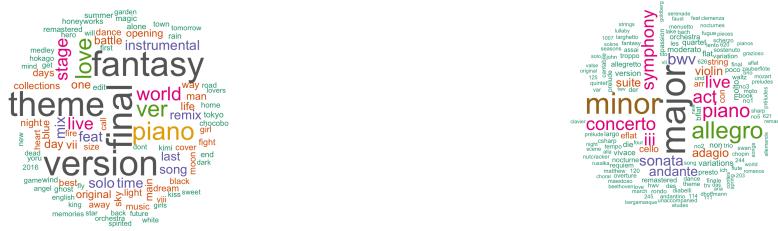


Figure 22: Anime music genre word-cloud Figure 23: Classical music genre word-cloud



Figure 24: Electronic music genre word-cloud

Figure 25: Rock music genre wordcloud

6.2 Idee per l'impiego e sviluppo di una nuova rete neurale

Vista la nuova quantità di dati si è dapprima pensato di addestrare una rete neurale con le sole dummy variables in modo da produrre in output la probabilità che una canzone appartenga ad uno specifico genere per poi usare l'output di questa prima rete neurale come input per la seconda, insieme a tutte le altre variabili numeriche viste precedentemente durante le analisi per produrre l'output finale della classificazione. Si è però successivamente scartata quest'idea vista la possibilità di trasmettere errori della prima rete neurale verso la seconda che si occupasse della vera e propria classificazione.

Si è quindi giunti a conclusione riguardo l'utilizzo di un'unica rete neurale più grande e più profonda di quella vista nel capitolo 5. La nuova rete neurale conta in input le 38 variabili (11 feature iniziali e 27 feature dummy generate come descritto nel sottocapitolo precedente), ha 2 strati nascosti composti da 27 e 18 neuroni rispettivamente e, infine, termina con 10 neuroni di output che permettono di classificare i diversi generi musicali previsti.

6.3 Nuovi risultati ottenuti

Un primo aspetto fondamentale per l'analisi dei risultati è sicuramente il tempo di calcolo, o meglio di addestramento, di questa rete neurale. Sono state impiegate diverse ore nonostante un processore di ultima generazione vista l'impossibilità di usare una GPU. Proprio in merito, per evitare eventuali rischi di fallimento, si è passati momentaneamente all'uso di python (e nello specifico con la libreria TensorFlow) proprio per valutarne l'efficacia e non investire eventuale tempo utile in un'idea fallimentare. Una volta raggiunti i risultati ottenuti si è addestrata la stessa rete neurale con R, ottenendo dunque i risultati che seguono.

Come evidente dalle tabelle 6 e 7 rappresentanti rispettivamente la matrice di confusione e le metriche ottenute, le performance sono di molto superiori rispetto ai precedenti modelli. La musica classica rimane il genere meglio classificato tra tutti, sia per la grande differenza con le caratteristiche musicali analizzate in fase di EDA sia per le ulteriori caratteristiche estratte che la differenziano rispetto agli altri generi riguardante i titoli.

Grazie a queste operazioni aggiuntive è stato possibile superare di gran lunga qualsiasi punteggio ottenuto dagli utenti su Kaggle relativamente a questo dataset.

pred	Alternative	Anime	Blues	Classical	Country	Electronic	Hip-Hop	Jazz	Rap	Rock
Alternative	878	53	33	12	32	63	54	45	23	55
Anime	41	1094	23	24	23	32	0	13	0	1
Blues	44	43	839	24	44	63	12	142	7	33
Classical	29	3	4	1189	2	8	0	14	0	0
Country	19	22	12	7	932	31	12	89	21	105
Electronic	70	43	0	9	67	955	21	65	12	9
Hip-Hop	78	14	21	17	40	2	727	25	251	76
Jazz	37	58	21	97	62	39	49	772	55	60
Rap	67	10	1	0	97	21	164	20	777	92
Rock	55	2	0	1	101	45	28	58	44	917

Table 6: Matrice di confusione per la seconda NN dove la diagonale viene evidenziata

Stats	Balanced Accuracy	AUROC	precision	recall	F1
Alternative	0.81654	0.75	0.6661608	0.7035256	0.6843336
Anime	0.90057	0.94	0.8152012	0.8745004	0.8438103
Blues	0.92189	0.87	0.8794549	0.6706635	0.7609977
Classical	0.92810	0.98	0.8615942	0.9519616	0.9045264
Country	0.81853	0.89	0.6657143	0.7456000	0.7033962
Electronic	0.86610	0.79	0.7585385	0.7633893	0.7609562
Hip-Hop	0.81776	0.82	0.6813496	0.5811351	0.6272649
Jazz	0.78931	0.78	0.6210780	0.6176000	0.6193341
Rap	0.80561	0.61	0.6529412	0.6220977	0.6371464
Rock	0.82516	0.74	0.6802671	0.7330136	0.7056560

Table 7: Metriche per la seconda NN dove sono evidenziati i risultati migliori ottenuti dal modello

7 Final considerations

Addestrati tutti i modelli è possibile osservarne i risultati raggiunti per determinare quali tra questi siano considerati migliori, sia in termini di tempo che di performance

7.1 Considerazioni temporali

Per questo progetto è stato utilizzato un dataset decisamente numeroso, con un numero di istanze pari a 50 000. Ogni modello è stato addestrato effettuando i calcoli sulla CPU, unico dispositivo hardware compatibile con il linguaggio di programmazione e scripting R. Quest'ultimo fattore ha quindi determinato in generale delle pessime performance temporali per la fase di addestramento che non sarebbero state tali nel qual caso si fosse utilizzato hardware in grado di effettuare i medesimi calcoli in parallelo.

Per ogni modello può essere effettuata una stima dei tempi di calcolo che è rappresentata in tabella 8.

	SVM	Neural Network	Neural Network w/ FE
Time (estimated)	~ 6 120s (1 hour 40 min)	~ 11 800s (3 hours)	~ 18 000s (5 hours)

Table 8: Timings for every model

Come evidente la macchina a vettore di supporto è sicuramente stata la più veloce nel classificare i vari generi musicali, seguita dalla rete neurale e infine dalla seconda rete neurale con l'aggiunta delle variabili estratte dai titoli delle tracce. L'andamento è quindi proporzionale rispetto alla complessità visto che le costruzioni delle reti neurali sono sicuramente più complesse della macchina a vettore di supporto, soprattutto nella seconda rete a causa di una presenza maggiore di attributi in input nonché di una profondità e ampiezza di layer nascosti più grandi rispetto alla rete precedente.

7.2 Confronto tra metriche

Dopo un'analisi dettagliata delle performance per ogni modello è importante effettuare un confronto generale tra questi per comprendere quale sia stato il

migliore nel classificare il task posto in essere. Tutte le metriche, come già sottolineato nella fase di test di ogni modello, sono state computate con lo stesso dataset di test dove le classi sono tutte bilanciate e i dati all'interno non hanno mai partecipato alla fase di addestramento dei modelli stessi. Si può quindi ottenere un'accurata, se pur non perfetta, rappresentazione di come ogni modello riesca a classificare dati mai visti in precedenza.

È per prima cosa possibile osservare l'accuratezza generale dei modelli vista la presenza di classi bilanciate e quindi perfettamente confrontabili. Non è ulteriormente presente nessuna classe di prevalenza rispetto alle altre. In figura 26 è possibile osservare come tra i tre modelli presentati, la rete neurale con le feature aggiuntive derivanti dalle Bag of Words riesce a ottenere un'accuratezza pari al 72,6%, circa il 19% in più della rete neurale precedente e della SVM.

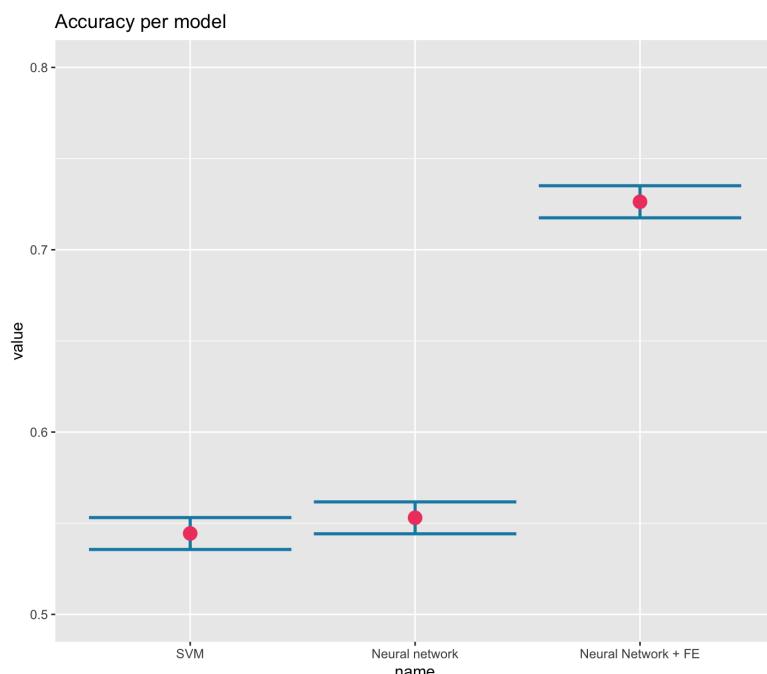


Figure 26: Accuracy for every model

Ulteriore analisi può essere effettuata osservando il grafico in figura 27 indicante i livelli di *Balanced accuracy* per ogni modello considerato (ovvero una media tra *Sensitivity* e *Specificity*). L'accuratezza bilanciata permette di considerare tutte le classi come equivalenti, a differenza del punteggio F1 che tende a prediligere una specifica classe considerata come positiva rispetto alla sua controparte negativa.

I risultati, anche qui prevedibili, sono in netto favore per la seconda rete neurale con le feature aggiuntive ricavate dalle BoW. È possibile notare soprattutto un significativo miglioramento per quanto riguarda le classi *Alternative*, *Blues*, *Country*, *Rap*, *Electronic*, *Hip Hop* e *Jazz*. Il genere id musica *Classic* invece conserva le sue buone performance già viste precedentemente.

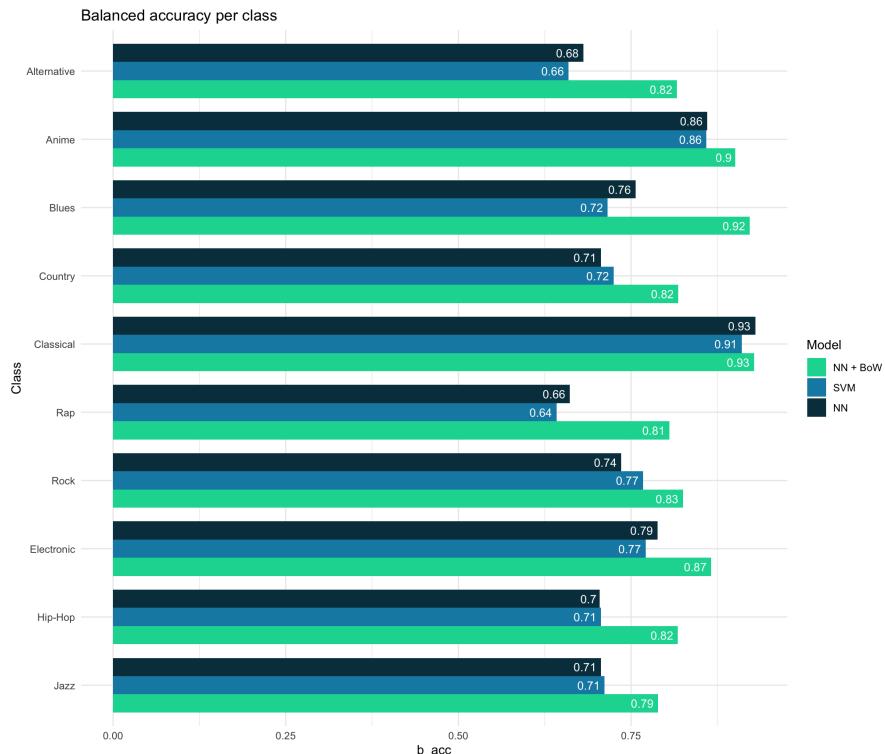


Figure 27: Balanced accuracy for every model

Medesimo ragionamento può essere osservato dalle curve ROC in figura 29 e dai valori di Area sotto la curva ROC in figura 28. Anche qui possiamo notare l'ottimo lavoro effettuato dal modello finale nel separare le diverse classi dove già la prima rete neurale iniziale otteneva punteggi migliori rispetto alla SVM. Si può quindi affermare che l'aggiunta di attributi relativi ai titoli delle tracce abbia aiutato nel separare le classi riducendo di molto la confusione tra queste data dai semplici dati originali del dataset.

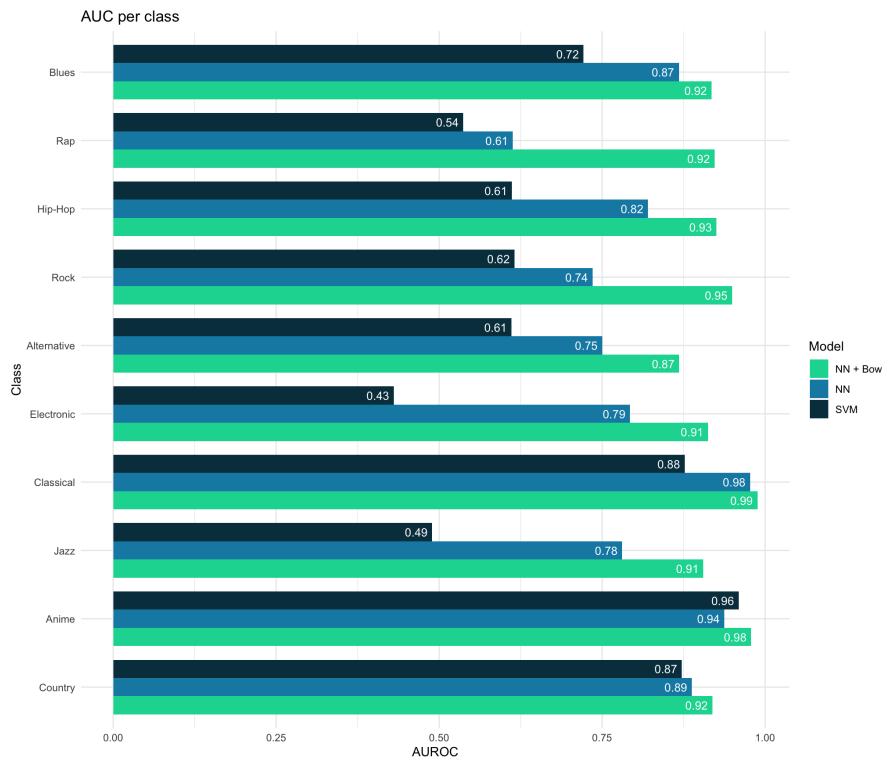


Figure 28: Area under the curve for the final NN model with new features applied

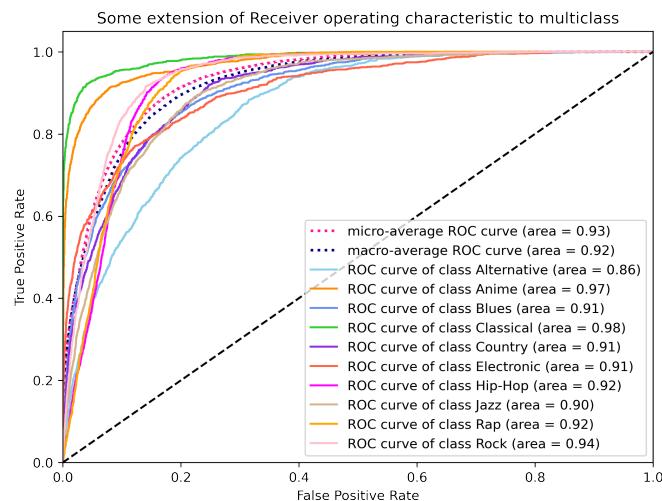


Figure 29: ROC curve for every class, regarding final NN model with new features applied