UNIVERSITÀ DI PISA

Data Mining Project Report

Bardazzi Carlo (603930)
Martini Virginia (667577)
Stira Alma (658699)

Summary

1. Introduction and Dataset Overview 2
2. Assessing data quality 3
2.2.1 Missing values 3
2.2.2 Duplicates ..... **Errore. Il segnalibro non è definito.**
2.2.3 Errors and semantics inconsistencies 3
2.2.4 Further analysis 4
3. Correlations and irrelevant attributes @alma @carlo 5
4. Outlier Detection and Dimensionality 7
4.1 Local Outlier Factor (LOF) 7
4.2 ABOD (Angle Based Outlier Degree) 8
4.3 LODA (Lightweight On-line Detector of Anomalies) 9
4.4 Isolation Forest 9
4.5 Conclusioni 10
5. Imbalanced Learning 11
6. Advanced Classification 12
6.1 Regressione Logistica 12
6.2 Support Vector Machines (SVM) 12
6.3 Reti Neurali 13
6.4 Metodi di Ensemble 13
6.4.1 Random Forest 13
6.4.2 Gradient Boosting 14
7. Advanced Regression 14
7.1 Logistic Regression 14
7.2 Gradient Boosting 16
7.3 Metriche di confronto 16
8. Time Series Analysis 17
8.1 Data Understanding and Preparation 17
8.2 Clustering 17
8.3.1 Partitional Clustering @TABELLE VIRGI 17
8.3.2 Hierarchical clustering 18
8.3.3 Features-based clustering 19
8.3 Classification 21
8.3.1 Pre-processing 21
8.3.2 Shapelets 21
8.3.3 Confronto tra Shapelets e Motifs 23
8.3.4 K-NN 23
8.3.5 Random Forest 24

8.3.6 Gradient Boosting 24
8.3.7 ROCKET 25
8.3.8 Conclusioni 25
8.4 Motifs and Discords 25
9. Explainability 26
9.1 Metodi Globali 27
9.1.1 Trepan (Tree-based Rule Extraction) 27
9.2 Metodi Locali 27
9.2.1 Valori SHAP (SHapley Additive exPlanations) 28
9.2.2 LIME (Local Interpretable Model-agnostic Explanations) @alma figure 28
9.3 Conclusioni 29
10. Conclusions ..... Errore. Il segnalibro non è definito.

1. Introduction and Dataset Overview

Il dataset "tracks.csv" è una collezione ricca e dettagliata di informazioni su 109,547 tracce musicali provenienti da Spotify. Il dataset contiene 34 colonne, ciascuna delle quali rappresenta un attributo specifico di una traccia musicale. Questi attributi forniscono una panoramica completa delle caratteristiche musicali e della popolarità delle tracce.

Le tracce musicali possono essere analizzate attraverso una varietà di caratteristiche per comprenderne meglio le proprietà e la popolarità. In questo progetto, ci proponiamo di esplorare e analizzare un dataset di tracce musicali per indagare vari attributi musicali e il loro impatto sulla popolarità delle tracce e sulla classificazione dei generi. Il dataset include caratteristiche come danceability, energy, key, loudness, e molte altre, fornendo una visione completa del profilo musicale di ciascuna traccia. Il dataset di riferimento è costituito da tracce di Spotify, ciascuna descritta da molteplici attributi.

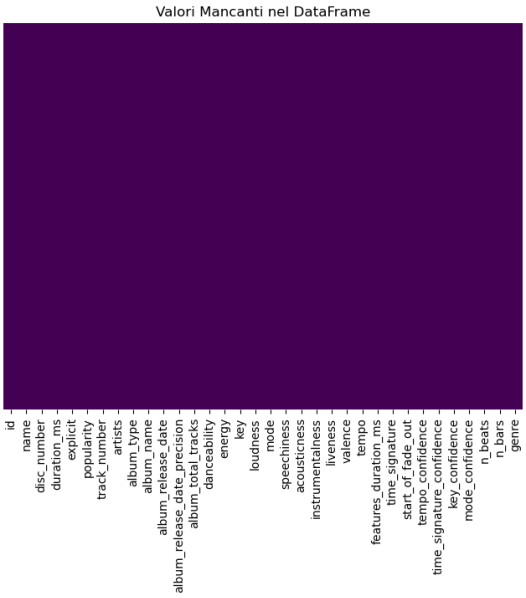
Prima di tutto, nella Sezione 2, vengono eseguite diverse operazioni di preparazione e comprensione dei dati, inclusa la trasformazione e l'eliminazione degli attributi, seguite da una suddivisione dei dati utile per esperimenti successivi. Successivamente, nella Sezione 4, viene effettuata una fase di rilevamento degli outlier, in cui vengono sfruttate anche tecniche di riduzione della dimensionalità. Poi, nella Sezione 5, viene testato uno scenario di classe sbilanciata, dove il dataset presenta un naturale squilibrio nelle classi. Si è deciso di testare due algoritmi per l'oversampling e uno per l'undersampling. Successivamente, viene eseguito un compito di classificazione per mostrare e confrontare i risultati. Nella Sezione 6, vengono eseguiti diversi algoritmi di classificazione multi-classe e regressione multipla delle caratteristiche. Per il compito di classificazione, si è deciso di classificare le tracce in diversi generi in base ai loro attributi.

Questa suddivisione dei dati ci consente di perseguire il compito principale del progetto: comprendere come varie caratteristiche musicali si correlano con la popolarità delle tracce e i generi, e come queste caratteristiche possono essere utilizzate per costruire modelli predittivi.

2. Assessing data quality

2.2.1 Missing values

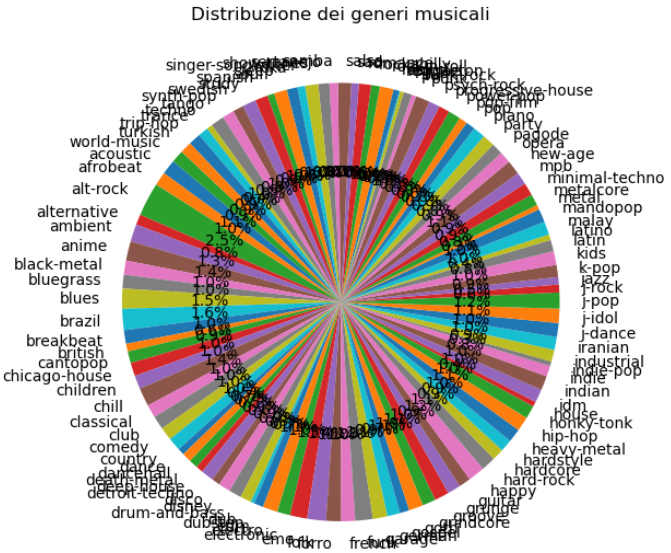
All'interno del nostro dataset, è stata condotta un'analisi approfondita per identificare eventuali valori mancanti. Questa verifica ha confermato l'assenza totale di valori mancanti nei dati a nostra disposizione

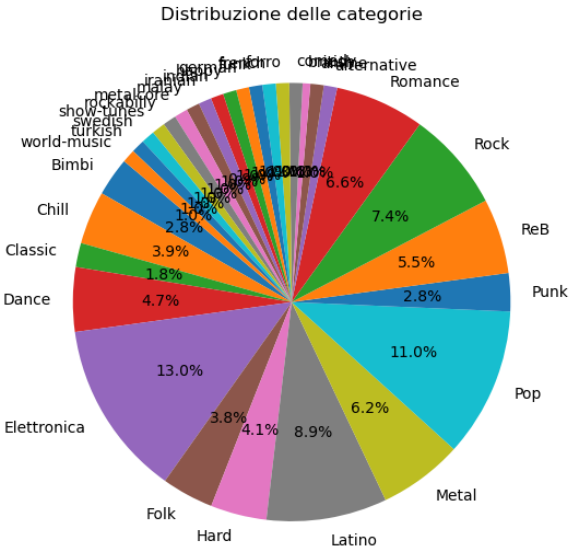


2.2.2 Errors and semantics inconsistencies

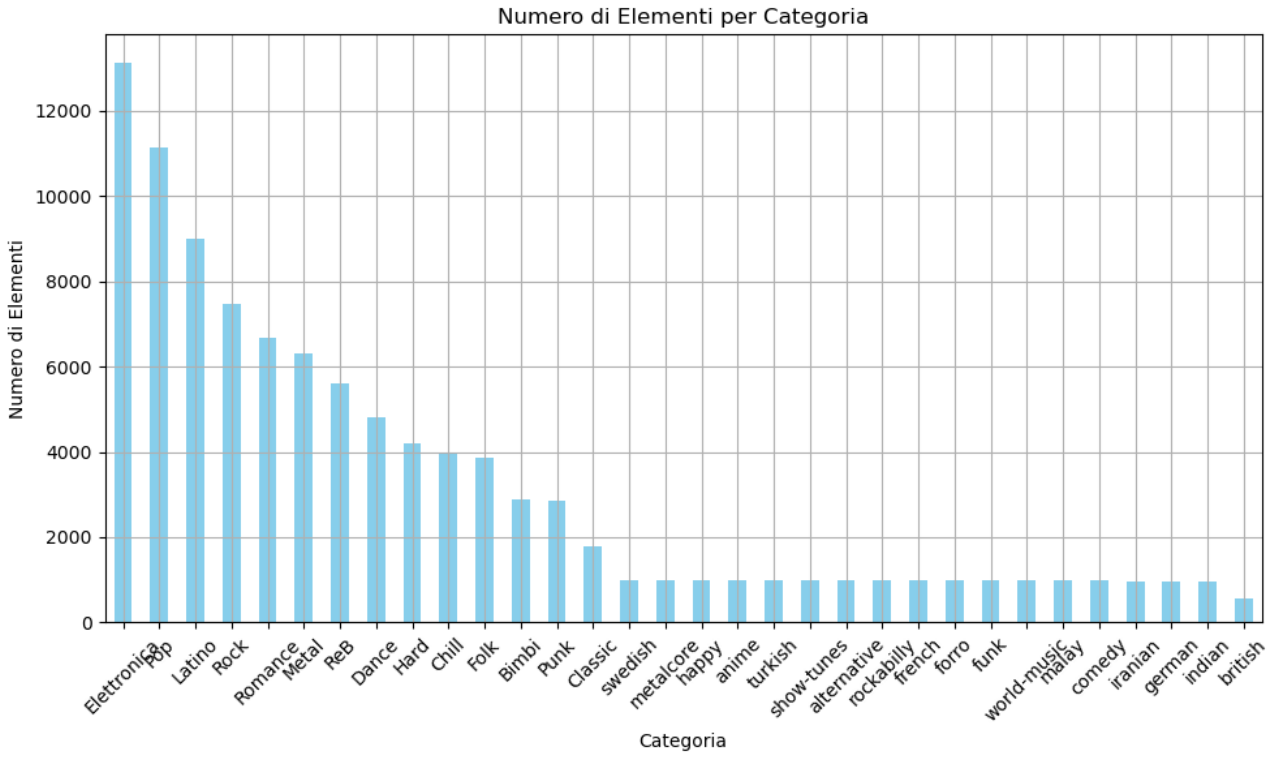
Durante l'analisi della variabile "firma del tempo", sono emerse due osservazioni degne di nota. In primo luogo, si è notata una discontinuità nei valori, con l'assenza del valore "2". Questa anomalia ha suggerito la possibilità di un errore nei dati, che implica un salto accidentale nei valori. Per ovviare a questa incongruenza, è stata sottratta un'unità a tutti i casi in cui i valori erano uguali a 3, 4 e 5. Inoltre, per quanto riguarda la variabile "time_signature", è stata rilevata l'incongruenza di valori nulli (0), che non ha alcun significato logico in quanto una firma temporale non può essere 0/4. Pertanto, per mantenere la coerenza dei dati, si è deciso di eliminare i valori nulli. Pertanto, per mantenere la coerenza e la validità dei dati, si è deciso di sostituire il valore 0 di "time_signature" con la modalità di altre canzoni dello stesso genere. Queste correzioni sono state implementate per garantire chiarezza e affidabilità dei dati, evitando ambiguità o interpretazioni errate nelle fasi successive dell'analisi.

Per quanto riguarda le colonne "key" e "mode", è stato deciso di unirle in un'unica colonna. Questa scelta è stata motivata da ragioni sia logiche che semantiche. La fusione delle due colonne permette di rappresentare in modo più coerente e significativo le caratteristiche tonali delle canzoni. Infatti, la combinazione di "key" (che indica la tonalità musicale) e "mode" (che distingue tra modalità maggiore e minore) fornisce una descrizione completa e univoca del contesto armonico di ogni brano musicale.Unendo queste due colonne, si ottiene una rappresentazione più accurata delle proprietà musicali





Main Genre	Subgenres
Elettronica	edm, idm, house, trance, deep-house, electro, electronic, chicago-house, minimal-techno, techno, drum-and-bass, progressive-house, breakbeat, synth-pop, garage, dub
Dance	dance, disco, dancehall, j-dance, club, party
Pop	pop, indie-pop, k-pop, j-pop, pop-film, power-pop, pop-rock, cantopop, mandopop, j-idol, afrobeat, hip-hop, trip-hop
Hard	hardcore, detroit-techno, dubstep, breakbeat, hardstyle, industrial
Rock	synth-pop, pop-rock, rock, punk-rock, rock-n-roll, alt-rock, j-rock, psych-rock, new-age, grunge
Metal	heavy-metal, metal, hard-rock, black-metal, metal-core, death-metal, grindcore, groove
Punk	emo, goth, punk, punk-rock
Latino	spanish, mpb, samba, salsa, pagode, latin, latino, reggae, reggaeton, tango, sertanejo, dub, brazil
Romance	sertanejo, romance, guitar, acoustic, songwriter, singer-songwriter, piano, sad, indie
Classic	classical, opera, piano
Chill	chill, study, sleep, ambient
ReB	r-n-b, jazz, soul, gospel, blues, ska
Folk	folk, blues, country, bluegrass, honky-tonk
Bimbi	children, kids, disney



2.2.3 Further analysis

Durante l'analisi, è stata notata la presenza dell'attributo 'processing', ma poiché non è direttamente descritto, si possono fare solo ipotesi basate sulle sue correlazioni con altre variabili e sulla sua distribuzione. La correlazione negativa con 'key' e quella positiva con 'mode' suggeriscono che "processing" potrebbe misurare come gli aspetti della tonalità e della modalità di una traccia influenzano o sono influenzati da altri processi o caratteristiche. Le correlazioni deboli con variabili come 'energy', 'danceability' e 'loudness' indicano che "processing" non dipende fortemente da queste caratteristiche percepite del suono. La distribuzione dei valori mostra una concentrazione attorno a un intervallo mediano, con alcuni valori estremi, il che potrebbe suggerire

che "processing" rappresenti una misura che varia tra le tracce e rifletta un'aggregazione di diverse caratteristiche musicali o di produzione. In conclusione, senza una descrizione esplicita, il significato esatto di "processing" rimane incerto, ma potrebbe essere interpretato come una misura o un indice del grado di difficoltà o qualità di elaborazione nelle tracce.

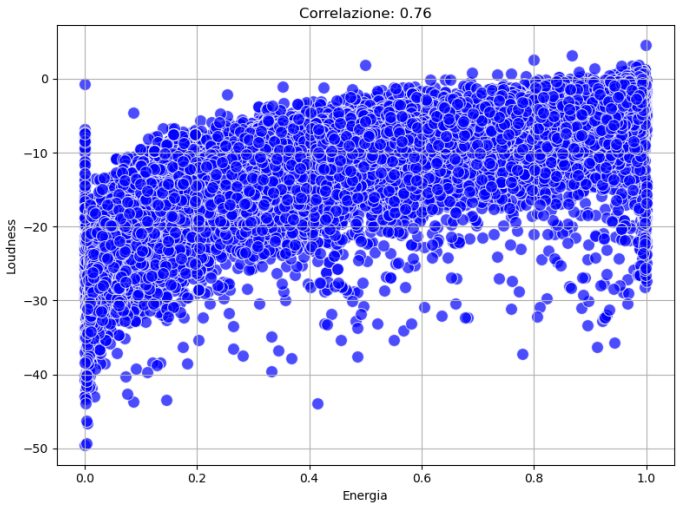
Inoltre, durante queste analisi, poiché i generi erano in totale 114 e molti di questi risultavano simili, si è deciso di unirli in delle macro-famiglie nel seguente modo:

1. Elettronica Soft: edm, idm, house, trance, deep-house, electro, electronic, chicago-house, minimal-techno, techno, drum-and-bass, progressive-house, breakbeat, synth-pop, garage, dub
2. Dance: dance, disco, dancehall, j-dance, club, party
3. Pop: pop, indie-pop, k-pop, j-pop, pop-film, power-pop, pop-rock, cantopop, mandopop, j-idol, afrobeat, hip-hop, trip-hop
4. Elettronica Hard: hardcore, detroit-techno, dubstep, breakbeat, hardstyle, industrial
5. Rock: synth-pop, pop-rock, rock, punk-rock, rock-n-roll, alt-rock, j-rock, psych-rock, new-age, grunge
6. Metal: heavy-metal, metal, hard-rock, black-metal, metal-core, death-metal, grindcore, groove
7. Altro: industrial, alternative, indian, iranian, anime, swedish, turkish, world-music, french, german, malay, british, show-tunes, comedy, happy
8. Punk: emo, goth, punk, punk-rock
9. Latino: spanish, mpb, samba, salsa, pagode, forro, latin, latino, reggae, reggaeton, tango, sertanejo, dub, brazil
10. Romance: sertanejo, romance, guitar, acoustic, songwriter, singer-songwriter, piano, sad, indie
11. Classic: classical, opera, piano
12. Chill: chill, study, sleep, ambient
13. R&B: r-n-b, jazz, soul, gospel, blues, ska
14. Folk: folk, blues, country, bluegrass, honky-tonk
15. Bimbi: children, kids, disney

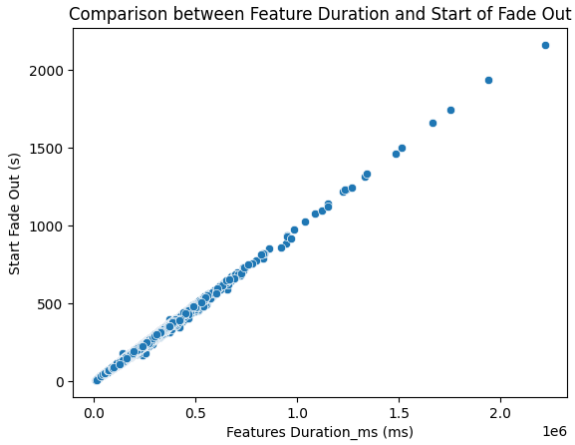
3. Correlations and irrelevant attributes @alma @carlo

L'analisi della correlazione è stata condotta per identificare le relazioni tra le diverse caratteristiche delle tracce musicali presenti nel dataset "tracks.csv". Utilizzando il metodo di Spearman, sono state calcolate le correlazioni tra le variabili numeriche. Di seguito, descriviamo i principali risultati emersi dall'analisi:

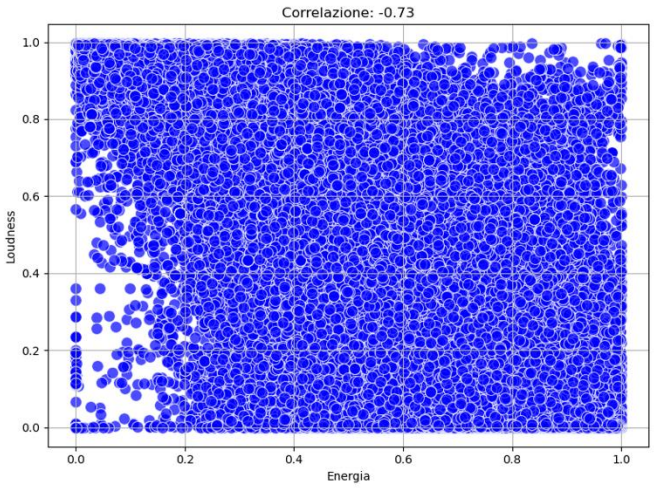
1. **Energia e Loudness (0.75)**
    - **Correlazione Positiva**: È stata osservata una forte correlazione positiva tra energy e loudness. Questo indica che le tracce con un'energia elevata tendono ad essere anche più rumorose. Pertanto, tracce musicali con alta energia, come quelle dei generi rock e dance, spesso hanno volumi più alti.



2. **Danceability e Valence (0.46)**
    - **Correlazione Positiva Moderata**: Esiste una correlazione positiva moderata tra danceability e valence. Questo suggerisce che tracce che sono facili da ballare tendono ad avere un tono più positivo o allegro. Generi come il pop e la dance music, che sono generalmente ballabili, spesso trasmettono sensazioni positive.
3. **Duration_ms e N_Beats e N_bars (0.80)**
    - **Correlazione Positiva**: La durata della traccia è positivamente correlata con il numero di battiti (n_beats). Tracce più lunghe naturalmente contengono più battiti, data la loro estensione temporale.
4. **Features_duration_ms e Start_of_fade_out (0.99)**
    - **Correlazione Positiva Forte**: Questo indica che, generalmente, più lunga è la traccia, più tardi inizia il fade out. La relazione è quasi lineare, suggerendo una connessione diretta tra la lunghezza della traccia e il punto in cui inizia a sfumare.



5. **Acousticness e Energia (-0.71)**
    - **Correlazione Negativa Forte**: È stata rilevata una forte correlazione negativa tra acousticness ed energy. Tracce acustiche tendono ad avere livelli di energia più bassi, il che è coerente con la natura più tranquilla e meno intensa della musica acustica.



L'analisi delle correlazioni tra le caratteristiche musicali delle tracce fornisce una comprensione dettagliata delle dinamiche interne del dataset. Questi risultati possono essere utilizzati per migliorare le raccomandazioni musicali, creare playlist personalizzate e sviluppare modelli predittivi più accurati per la popolarità e la classificazione dei generi musicali.

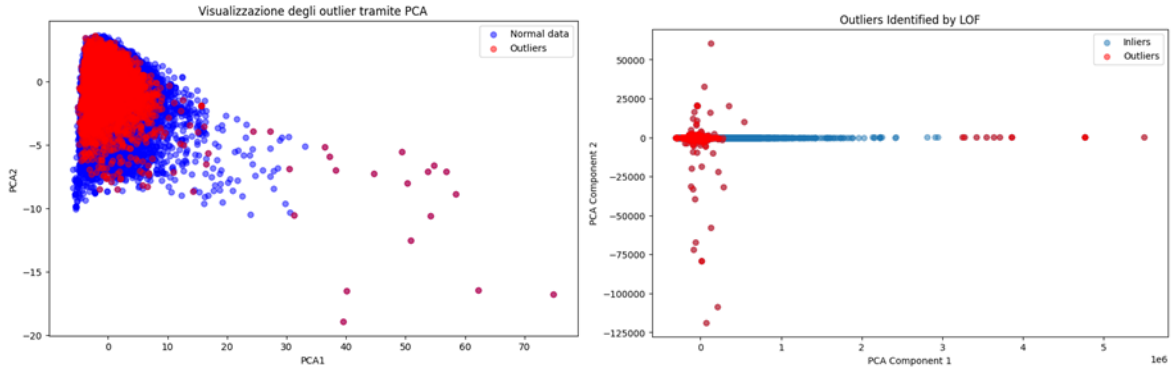
4. Outlier Detection and Dimensionality

In questa fase siamo andati alla ricerca degli outliers, vale a dire quei valori che possono influenzare negativamente i risultati dei modelli e, pertanto, è essenziale identificarli e gestirli correttamente. In particolare, siamo andati alla ricerca dell'1% dei punti più estremi, identificando così quelle che sono le anomalie più significative, filtrando il rumore e concentrandosi sui punti che potrebbero avere un impatto maggiore sul sistema o sul modello. Abbiamo utilizzato questo approccio sia per ridurre il carico computazionale e di memoria, migliorando l'efficienza delle analisi sia perché in questo modo è più probabile di identificare anomalie reali i, riducendo i falsi positivi e migliorando la precisione delle successive analisi.

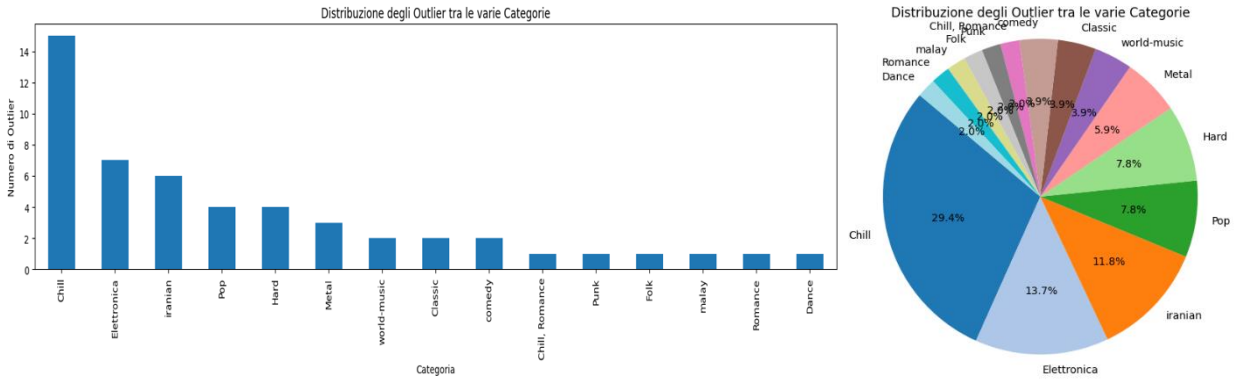
Durante la nostra analisi non abbiamo preso in considerazione le confidence (tempo_confidence, time_signature_confidence, key_confidence, mode_confidence, popularity_confidence) perche given the qualitative nature of this attribute, it was determined that values which might otherwise be considered outliers should not be excessively categorized as such. This decision was motivated by the intention to preserve vital information regarding the quality of the measurement of a track's popularity, whilst maintaining an informative and contextually precise approach.

4.1 Local Outlier Factor (LOF)

In prima battuta per rilevare gli outliers, abbiamo utilizzato il modello LOF, il quale si basa sulla densità locale di un punto rispetto ai suoi vicini, classificando come outliers quei punti che hanno una densità significativamente inferiore rispetto agli altri. Una volta rilevati gli outliers, 4479 punti, abbiamo applicato la PCA per ridurre la dimensionalità dei dati e facilitare la loro visualizzazione in uno spazio bidimensionale, questa ha prodotto un grafico dove i punti blu rappresentano i dati normali e i punti rossi rappresentano gli outliers. Si osserva che gli outliers tendono a deviare significativamente dalla massa centrale dei dati, confermando la loro natura anomala.



I grafici seguenti grafici, invece, illustrano la distribuzione degli outliers tra diverse categorie e mostrano come questi punti anomali si distribuiscono nel dataset.

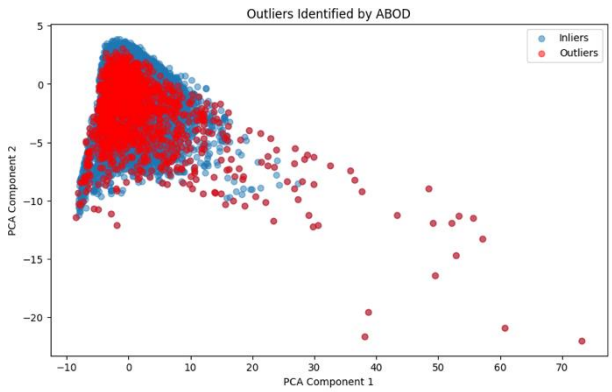


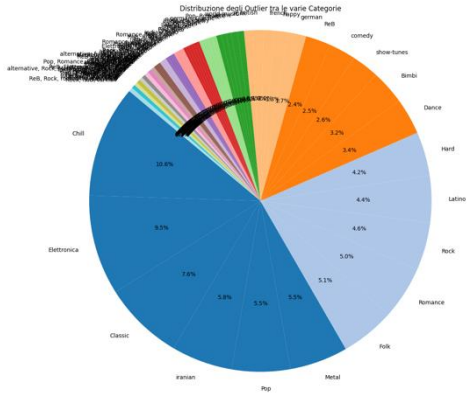
Qui è evidente che alcune categorie, come "Chill" e "Elettronica", presentano un numero significativamente maggiore di outliers rispetto ad altre. Questo potrebbe indicare che in queste categorie ci sono comportamenti o dati anomali più frequenti.

4.2 ABOD (Angle Based Outlier Degree)

Il secondo algoritmo utilizzato è ABOD, questo è un approccio pensato specificamente per dataset ad alta dimensionalità. Questo metodo sfrutta gli angoli per calcolare l'outlier score, poiché questi sono più stabili rispetto alle distanze in spazi multidimensionali. I punti considerati come outlier sono quelli rispetto ai quali il resto dei dati si posiziona in direzioni simili. In altre parole, un punto viene considerato normale se la varianza degli angoli è grande, indicando che ci sono punti distribuiti in ogni direzione. Gli outliers, quindi, tendono a trovarsi ai bordi delle distribuzioni. Il modello misura la varianza dello spettro degli angoli formati dal punto in esame rispetto a tutti gli altri punti.

I dati per questo approccio sono stati inizialmente standardizzati utilizzando lo StandardScaler. Mentre la PCA è stata utilizzata per ridurre la dimensionalità dei dati e facilitare la visualizzazione degli outliers in uno spazio bidimensionale.

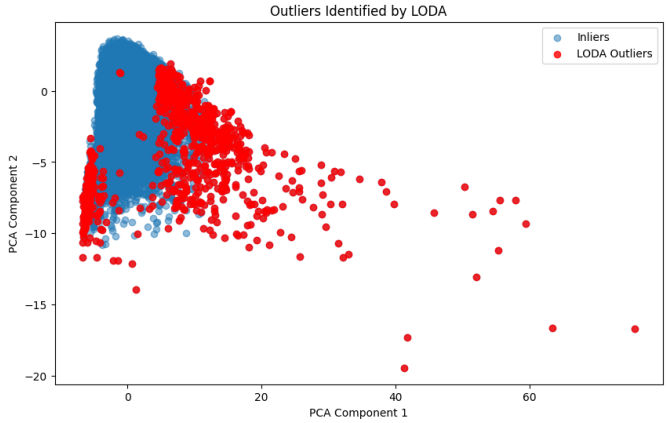


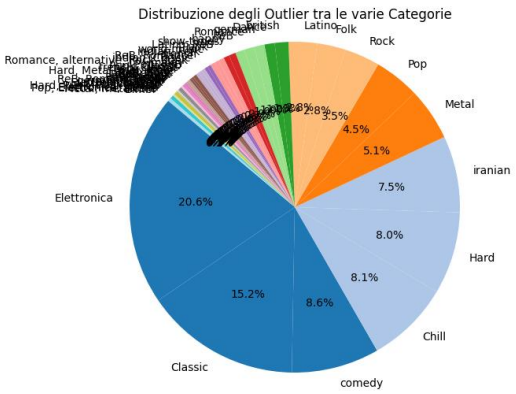


Mentre LOF identificava un numero significativo di outliers per "Chill", "Elettronica" e "Iranian", ABOD evidenzia un maggior numero nelle categorie "Elettronica", "Classic" e "Rock".

4.3 LODA (Lightweight On-line Detector of Anomalies)

Anche LODA è un algoritmo progettato per rilevare anomalie in dataset di grandi dimensioni, questo si basa sull'utilizzo di proiezioni casuali dei dati in spazi di dimensioni ridotte, combinando poi i risultati di queste proiezioni per calcolare un punteggio di anomalia. Anche in questo caso abbiamo utilizzato StandardScaler e successivamente la PCA per la visualizzazione.

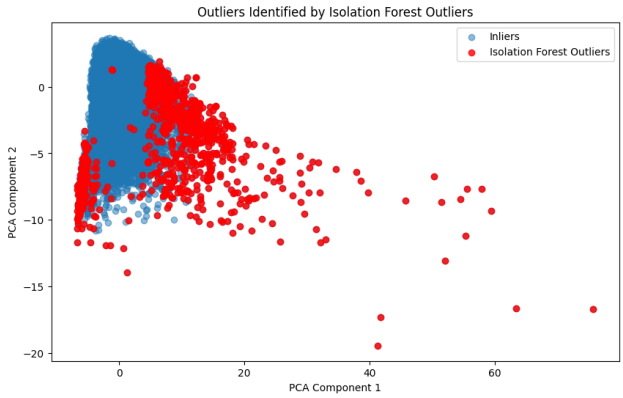


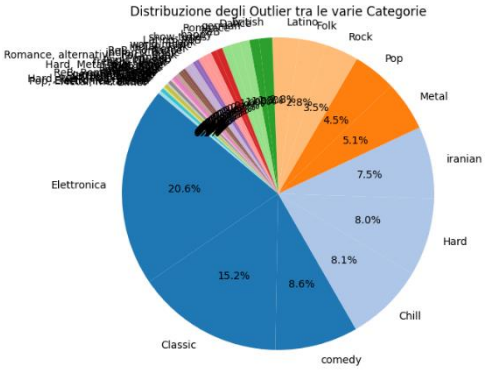


Con questa analisi le categorie con un numero elevato di outliers risultano essere: "Elettronica" e "Classic".

4.4 Isolation Forest

Isolation Forest è l'ultimo degli algoritmi utilizzati per l'individuazione degli outliers nel nostro dataset. Questo algoritmo appartiene alla famiglia degli approcci basati su modelli, il che significa che costruisce un modello per classificare i dati e può essere utilizzato per valutare anche nuovi record successivi. L'idea di base di Isolation Forest è quella di costruire una foresta di alberi di decisione. Durante la costruzione degli alberi, si scelgono casualmente le dimensioni e i valori rispetto ai quali dividere il dataset. Questo processo continua fino a raggiungere un numero minimo di record nei nodi foglia o una certa profondità dell'albero. I punti che vengono frequentemente isolati in nodi foglia a livelli superiori degli alberi sono considerati outliers. Per ogni punto nel dataset, viene calcolato un punteggio basato su tutti gli alberi costruiti. Un punteggio più vicino a 1 indica una maggiore probabilità che il punto sia un outlier, poiché è stato isolato più rapidamente rispetto agli altri punti.

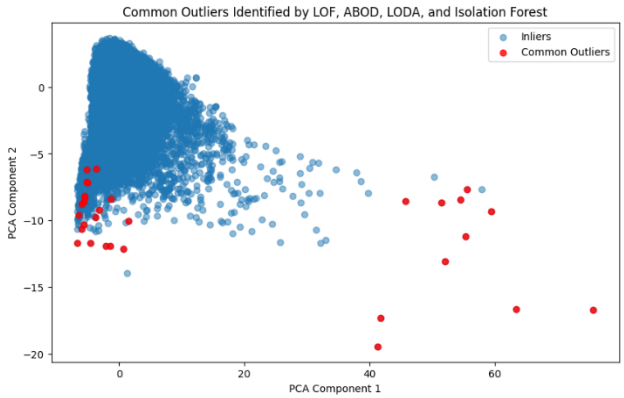




Questa analisi ha confermato ulteriormente che le categorie "Elettronica", "Classic" e "Comedy" presentano un numero significativo di anomalie.

4.5 Conclusioni

@carlo



Per confrontare i risultati ottenuti dai vari algoritmi di rilevamento degli outlier, abbiamo identificato gli outlier comuni individuati dai metodi Local Outlier Factor (LOF), Angle-Based Outlier Detection (ABOD), Lightweight Online Detector of Anomalies (LODA) e Isolation Forest. La nostra analisi iniziale ha rivelato che combinando tutti gli outlier rilevati da ciascuno di questi metodi, il numero totale di outlier risultava essere eccessivamente elevato. Di conseguenza, abbiamo deciso di adottare un criterio più restrittivo, considerando solo gli outlier che venivano individuati da almeno quattro dei suddetti algoritmi.

Una volta identificati gli outlier comuni secondo questo criterio, abbiamo sperimentato diverse strategie per gestire questi valori anomali. Le strategie testate includevano la rimozione degli outlier, la loro sostituzione con il valore medio, con il valore mediano o con il valore più frequente (moda). Per determinare la migliore strategia da adottare, abbiamo valutato non solo la quantità di outlier rilevati nel nuovo dataset risultante, ma anche come queste strategie influivano sulle distribuzioni dei dati.

Dall'analisi dei risultati, è emerso che la sostituzione degli outlier con il valore medio rappresentava la soluzione ottimale. Questa metodologia non solo riduceva in modo significativo il numero di outlier nel dataset finale, ma manteneva anche distribuzioni dei dati più omogenee e coerenti con quelle originarie. Pertanto, abbiamo deciso di adottare la sostituzione con la media come piano d'azione principale per la gestione degli outlier nel nostro processo di analisi dei dati.

Questa scelta metodologica ha permesso di preservare l'integrità e la qualità dei dati, facilitando al contempo analisi più accurate e affidabili. In sintesi, la sostituzione con il valore medio si è rivelata la strategia più efficace per gestire gli outlier, garantendo una riduzione degli effetti negativi degli stessi e migliorando la consistenza delle distribuzioni dei dati nel dataset finale. Data la performance ottenuta dall'unione dei diversi algoritmi riteniamo che l'identificazione di outliers comuni tra più metodi può aumentare la robustezza dell'analisi e fornire una base più solida per prendere decisioni informate e ottimizzare i modelli di machine learning.

5. Imbalanced Learning

In questo studio sull'imbalanced learning, è stato affrontato il problema del dataset sbilanciato utilizzando diverse tecniche di bilanciamento, focalizzandosi principalmente su undersampling e oversampling con i modelli Decision Tree e K-Nearest Neighbors (KNN).

Nella sezione sull'undersampling, sono stati esplorati due approcci principali: RandomUnderSampler (RUS) e ClusterCentroids (CC). Con entrambe le tecniche, sia Decision Tree che KNN hanno mostrato prestazioni generalmente scarse. Il RUS ha migliorato leggermente la recall, ma con una precisione ancora bassa, mentre il CC ha mantenuto una distribuzione dei dati più fedele ma con risultati complessivamente insoddisfacenti.

Undersampling Techniques	KNN		Decision Tree	
RandomUnderSampler	**Precision:**	0.06	**Precision:**	0.20
	Recall:	0.07	**Recall:**	0.27
	F1-Score:	0.06	**F1-Score:**	0.20
ClusterCentroids	**Precision:**	0.03	**Precision:**	0.12
	Recall:	0.04	**Recall:**	0.14
	F1-Score:	0.03	**F1-Score:**	0.09

Passando all'oversampling, sono state valutate tecniche come SMOTE e RandomOverSampler. SMOTE ha prodotto miglioramenti significativi per il Decision Tree, aumentando la precisione, il recall e l'accuratezza complessiva. Tuttavia, il KNN ha beneficiato meno delle tecniche di oversampling, mostrando solo lievi miglioramenti.

Overrsampling Techniques	KNN		Decision Tree	
RandomOverSampler	**Precision:**	0.08	**Precision:**	0.22
	Recall:	0.10	**Recall:**	0.25
	F1-Score:	0.08	**F1-Score:**	0.23
SMOTE	**Precision:**	0.09	**Precision:**	0.23
	Recall:	0.11	**Recall:**	0.26
	F1-Score:	0.09	**F1-Score:**	0.24

L'ADASYN è stata esplorata ma ha fallito nel generare campioni sintetici efficaci a causa della distribuzione estremamente sbilanciata delle classi nel dataset musicale in esame.

Le conclusioni generali hanno evidenziato che il bilanciamento delle classi è cruciale per migliorare le performance dei modelli di machine learning, soprattutto in contesti con dataset fortemente sbilanciati. SMOTE è emerso come la tecnica più efficace per questo specifico dataset, migliorando significativamente le performance dei modelli, soprattutto per il Decision Tree. In definitiva, l'analisi ha sottolineato l'importanza di bilanciare accuratamente i dataset per evitare perdite di informazioni significative (come nel caso dell'undersampling) e migliorare la capacità predittiva dei modelli di machine learning.

6. Advanced Classification

Per questa parte dell'analisi, affronteremo il task di classificazione precedentemente definito utilizzando diversi metodi di classificazione avanzati: Regressione Logistica, Support Vector Machines (SVM), Reti Neurali, Metodi di Ensemble e Gradient Boosting Machines. Per ogni metodo, verranno eseguite fasi di tuning degli iperparametri e verranno valutate le performance utilizzando le metriche di valutazione appropriate.

6.1 Regressione Logistica

Abbiamo applicato la regressione logistica al nostro dataset di classificazione delle categorie musicali. Dopo aver diviso il dataset in set di addestramento e test, abbiamo utilizzato il metodo SMOTE per bilanciare le classi nel set di addestramento. Successivamente, abbiamo eseguito una ricerca su griglia per ottimizzare gli iperparametri del modello di regressione logistica, trovando i migliori parametri: C=10, penalty='l2' e solver='liblinear'.

Performance del Modello:

- **Accuratezza**: 0.20
- **Precision media (macro)**: 0.21
- **Recall medio (macro)**: 0.32
- **F1-Score medio (macro)**: 0.20
- **ROC AUC Score**: 84%

Nonostante il buon ROC AUC Score che evidenzia una buona capacità di discriminazione, la bassa accuratezza e i valori di precisione, recall e F1-score indicano che il modello di regressione logistica non è sufficientemente efficace nel classificare correttamente tutte le categorie musicali. Questo potrebbe essere dovuto all'alta complessità del dataset e alla natura sbilanciata delle classi.

6.2 Support Vector Machines (SVM)

In questo studio, è stato applicato un modello SVM (Support Vector Machine) al dataset di classificazione delle categorie musicali. Dopo aver eseguito una ricerca su griglia per ottimizzare gli iperparametri, i migliori parametri trovati sono stati C=1, gamma='scale' e kernel='linear'.

Per ottimizzare il tempo di calcolo, sono state apportate le seguenti modifiche:

- Riduzione della Griglia dei Parametri: È stato ridotto il numero di parametri per kernel e gamma per accelerare il processo di ricerca.
- Riduzione dei Fold della Cross-Validazione: Il valore di cv è stato cambiato da 5 a 3 per velocizzare la validazione incrociata.
- Parallelizzazione: È stato aggiunto n_jobs=-1 a GridSearchCV per sfruttare tutti i core disponibili della CPU.

Performance del Modello

- **Accuratezza**: 80%
- **Precisione Media**: 78%
- **Recall Medio**: 81%
- **F1-Score Medio**: 79%

Il modello SVM ha dimostrato di essere molto efficace nel classificare la maggior parte delle categorie musicali nel dataset. Tuttavia, alcune classi richiedono ulteriori miglioramenti. Potrebbe essere utile esplorare tecniche

avanzate di pre-processing dei dati o bilanciamento delle classi, o provare altre architetture di modelli più complessi per migliorare ulteriormente le performance.

I risultati complessivi suggeriscono che l'SVM con kernel lineare è una scelta valida per questo tipo di classificazione, offrendo un buon equilibrio tra precisione e recall per la maggior parte delle classi.

6.3 Reti Neurali

Le reti neurali offrono grande flessibilità e sono particolarmente potenti per problemi complessi. Anche se richiedono più tempo per l'addestramento e una fase di tuning degli iperparametri più complessa, hanno mostrato buone performance in termini di F1-score. I migliori parametri trovati sono stati: learning_rate_init=0.001, hidden_layer_sizes=(50,), e alpha=0.01.

Performance del Modello

- **Accuratezza**: 0.29
- **Precisione, Recall e F1-Score:** Le metriche variano significativamente tra le categorie.
- **ROC AUC Score**: Un AUC medio di 0.89 suggerisce una buona capacità di distinguere tra le classi.

Il modello di reti neurali ha mostrato un buon potenziale nel classificare le categorie musicali, con risultati promettenti in diverse metriche di valutazione.

6.4 Metodi di Ensemble

Gli ensemble methods come Random Forest e Gradient Boosting combinano più deboli apprenditori per migliorare le performance.

Entrambi i metodi di ensemble hanno mostrato buone performance e robustezza. In particolare, il Gradient Boosting si è rivelato molto efficace nel migliorare le performance di classificazione. Questi metodi combinano le previsioni di molti deboli apprenditori per produrre un modello robusto e accurato.

6.4.1 Random Forest

Il modello Random Forest, ottimizzato con i parametri trovati tramite GridSearchCV {'n_estimators': 200, 'min_samples_split': 5, 'min_samples_leaf': 1, 'max_depth': 30, 'bootstrap': False, ha mostrato prestazioni eccellenti nella classificazione delle categorie musicali. Ecco un'analisi dettagliata delle metriche di performance:

Metriche di Performance:

- **Accuracy**: 0.85
- **Precisione**: 0.83 (macro)
- **Recall**: 0.90 (macro)
- **F1-Score**: 0.85 (macro)
- **ROC AUC Score**: 0.99

Pertanto, il modello si è dimostrato altamente efficace per questo task di classificazione delle categorie musicali. La combinazione di alta precisione, recall e AUC suggerisce che il modello è ben bilanciato e adatto per applicazioni pratiche. La combinazione di robustezza, capacità di gestione delle feature, mitigazione dell'overfitting, e l'uso efficace di tecniche di bilanciamento e ingegneria delle feature ha reso il modello Random Forest particolarmente efficace per la classificazione delle categorie musicali.

6.4.2 Gradient Boosting

Il modello di Gradient Boosting ha mostrato performance notevoli sui dati di test. I migliori parametri trovati sono stati: {'subsample': 0.8, 'n_estimators': 200, 'min_samples_split': 10, 'max_depth': 5, 'learning_rate': 0.1} .

Performance del Modello Gradient Boosting:

- Precision: 0.84
- Recall: 0.87
- F1-Score: 0.85
- ROC AUC Score: 1.00

Il modello Gradient Boosting ha dimostrato di essere estremamente efficace per il task di classificazione delle categorie musicali, ottenendo performance eccellenti su diverse metriche di valutazione. Con un'accuratezza complessiva dell'86%, una precisione del 84%, un recall del 87% e un F1-Score del 85%, il modello ha mostrato una capacità di classificazione robusta e affidabile. Inoltre, il ROC AUC Score perfetto di 1.00 indica che il modello è altamente competente nel distinguere tra le varie classi.

In sintesi, il Gradient Boosting si è rivelato un metodo potente e versatile per la classificazione multiclasse, capace di gestire dataset complessi e fornire risultati accurati e coerenti.

6.1 Conclusioni

I modelli di ensemble, in particolare Random Forest e Gradient Boosting, hanno mostrato le migliori performance per il nostro task di classificazione delle categorie musicali. La loro capacità di combinare le previsioni di più deboli apprenditori ha portato a risultati robusti e accurati. Le reti neurali, sebbene richiedano più risorse computazionali e tempo di addestramento, hanno mostrato un buon potenziale e potrebbero beneficiare di ulteriori miglioramenti e tuning. La regressione logistica, pur essendo più semplice da implementare, ha mostrato limiti nelle performance a causa della complessità del dataset. Infine, l'SVM si è rivelato un modello valido con buone performance, particolarmente efficace per alcune classi.

7. Advanced Regression

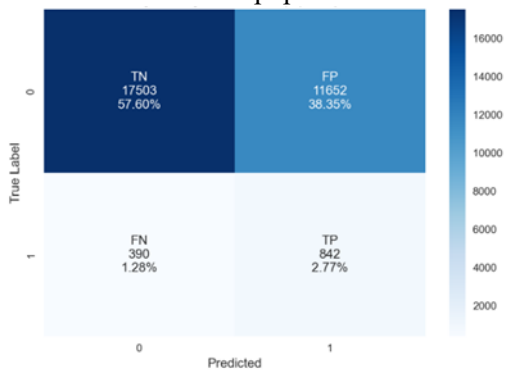
In questa parte andremo a discutere e ad analizzare le regression task. Come obiettivo di tali task ci siamo prefissati di capire se un brano fosse popolare. Per prima cosa abbiamo binarizzato la nostra variabile mettendo a 1 qualora la popolarità avesse un valore superiore a 70, 0 altrimenti. Come approcci abbiamo usato la logistic regression e il gradient boosting.
Come attributi per effettuare l'analisi abbiamo preso in considerazione tutti quelli che esprimono una qualità del brano e genere, chiave/mode e time signature su cui abbiamo fatto l'operazione di one-hot encoding.

7.1 Logistic Regression

Dopo aver diviso il dataset in set di addestramento e test, abbiamo eseguito una ricerca su griglia per ottimizzare gli iperparametri del modello, C=10, solver='lbfgs', class_weight='balanced'.

Abbiamo preso *C=10* in modo tale che il modello si adattasse meglio ai dati di training poiché non ci aspettavamo molto rumore nel dataset. Il solver "*lbfgs*" è stato scelto a causa del grande numero di variabili prese in considerazione per l'analisi iniziale e per le elevate dimensioni del dataset. Inoltre, abbiamo messo la class weight a "*balanced"* per la grande disparità del numero di elementi nella popolarità.

	precision	recall	f1-score	support
0	0.98	0.60	0.74	29155
1	0.07	0.68	0.12	1232
accuracy			0.60	30387
macro avg	0.52	0.64	0.43	30387
weighted avg	0.94	0.60	0.72	30387

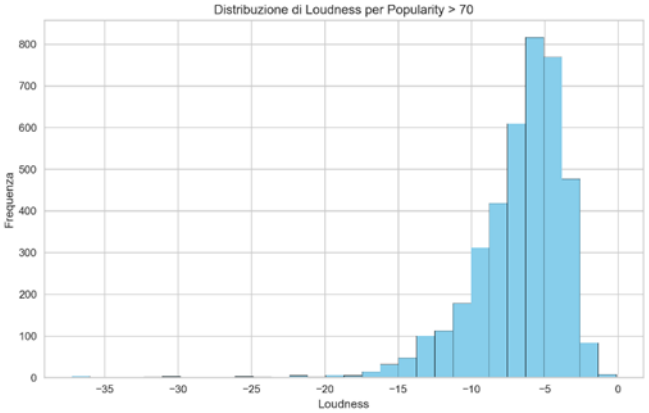


Dopo una prima analisi siamo andati a controllare i coefficienti di regressione per capire quali fossero le feature che aiutavano a definire meglio la popolarità.

```
Nomi delle features con coefficienti positivi: 'explicit', 'danceability', 'loudness', 'valence', 'categoria_Chill', 'categoria
_Dance', 'categoria_Elettronica', 'categoria_Folk', 'categoria_Latino', 'categoria_Metal', 'categoria_Pop', 'categoria_Punk',
'categoria_ReB', 'categoria_Rock', 'categoria_Romance', 'categoria_alternative', 'categoria_british', 'categoria_french', 'cate
goria_funk', 'categoria_german', 'categoria_indian', 'categoria_rockabilly', 'categoria_show-tunes', 'categoria_swedish', 'time
_signature_1', 'time_signature_3', 'time_signature_4', 'chord_type_n_1', 'chord_type_n_2', 'chord_type_n_5', 'chord_type_n_7',
'chord_type_n_8', 'chord_type_n_9', 'chord_type_n_12', 'chord_type_n_16', 'chord_type_n_17', 'chord_type_n_18', 'chord_type_n_2
0', 'chord_type_n_21', 'chord_type_n_22'
```

Dopo varie prove abbiamo osservato che le principali features che aiutano a identificare la popolarità sono: explicit, loudness, time signature e key/mode. Inoltre, si può notare come i generi più popolari sono: Elettronica, Folk, Metal, Pop, R&B, Rock, Romance e Alternative.

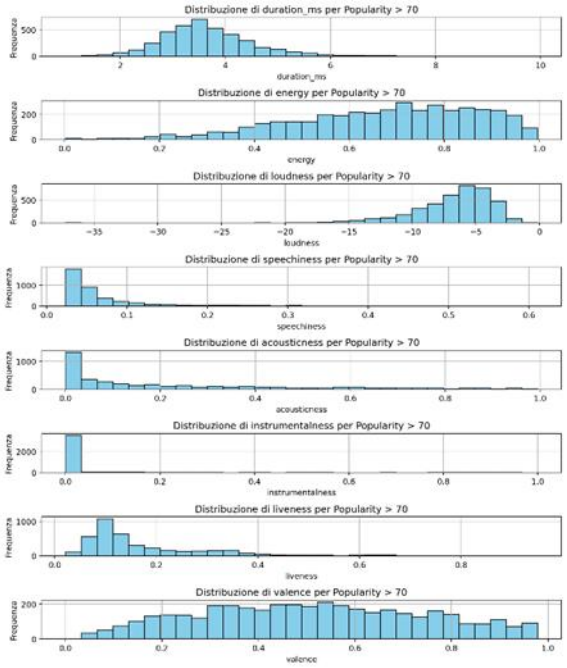
	precision	recall	f1-score	support
0	0.99	0.64	0.77	29155
1	0.08	0.79	0.15	1232
accuracy			0.64	30387
macro avg	0.54	0.71	0.46	30387
weighted avg	0.95	0.64	0.75	30387



7.2 Gradient Boosting

Dopo aver diviso il dataset in set di addestramento e test, abbiamo eseguito una ricerca su griglia per ottimizzare gli iperparametri del modello. I migliori parametri trovati sono stati: {'n_estimators': 200, 'min_samples_split': 10, 'max_depth': 7, 'learning_rate': 0.1}.
Per la scelta dei parametri del modello ci siamo concentrati principalmente sull'ottenere un modello lievemente più lento ma che andasse più nel dettaglio in modo tale da ottenere risultati migliori e senza preoccuparsi troppo del rumore dato le modifiche effettuate precedentemente al dataset che lo dovrebbero aver ridotto considerevolmente.
Anche in questo caso, inizialmente abbiamo usato le stesse feature usate per la Logistic Regression andando poi a fare una scrematura basandosi sulla feature importance del modello. Da questa analisi, è emerso che a differenza della Logistic Regression gli attributi che maggiormente aiutano a capire la popolarità sono attributi che esprimono una caratteristica della canzone, tra cui: Duration_ms, Energy, Loudness, Speechiness, Acousticness, Instrumentalness, Liveness, Valence.

	precision	recall	f1-score	support
0	0.96	1.00	0.98	29155
1	0.00	0.00	0.00	1232
accuracy			0.96	30387
macro avg	0.48	0.50	0.49	30387
weighted avg	0.92	0.96	0.94	30387



7.3 Metriche di confronto

Metriche	Logistic Regression	Gradient Boosting
Mean Absolute Error (MAE)	0.35590877677954386	0.040675288774805016
Mean Squared Error (MSE)	0.35590877677954386	0.040675288774805016
Root Mean Squared Error (RMSE)	0.5965809054768212	0.20168115622141056
R-squared (R^2)	-8.14935747026083	-0.04564085374409488

Il primo set di metriche indica che il modello ha prestazioni molto scadenti. Il MAE, il MSE e il RMSE sono molto alti, il che suggerisce che il modello predice in modo significativamente errato rispetto ai valori osservati. Inoltre, il valore molto negativo di R^2 conferma che il modello è inefficace nel fare predizioni migliori rispetto a una stima media semplice.

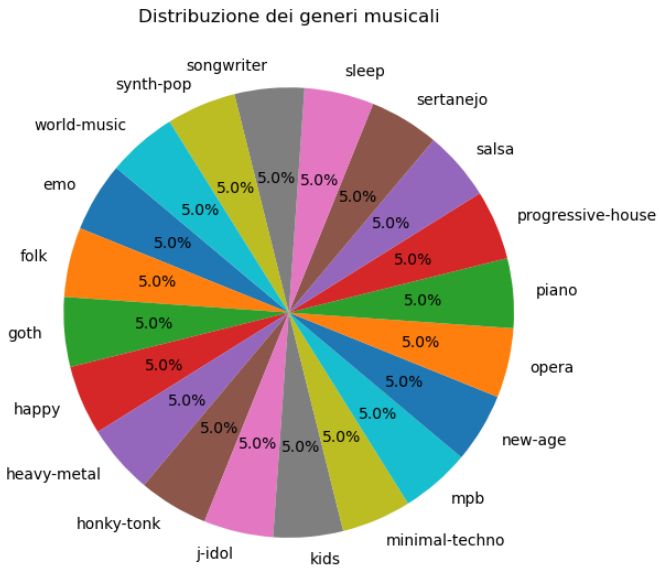
Nel secondo set di metriche, il modello mostra un MAE, un MSE e un RMSE molto bassi, indicando che le predizioni sono molto vicine ai valori osservati. Tuttavia, il valore negativo di R^2 (-0.0456) suggerisce che il

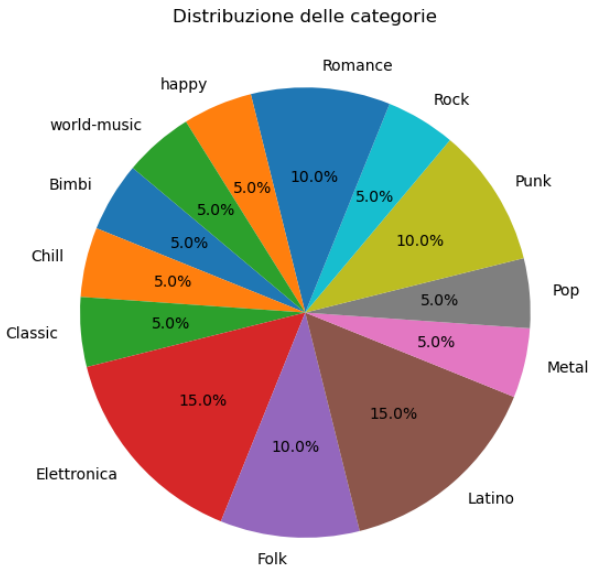
modello non riesce a spiegare la variazione nei dati meglio di un semplice modello che prevede la media dei dati. Potrebbe essere necessario esaminare più da vicino il modello per migliorare la sua capacità predittiva.

8. Time Series Analysis

8.1 Data Understanding and Preparation

All'interno delle time series sono identificati 20 generi, diversamente dal dataset tabulare. Poiché abbiamo ridefinito i generi nel dataset, abbiamo esaminato anche qui le eventuali modifiche risultanti. Dopo aver convertito i dati per uniformare la rappresentazione dei generi, abbiamo osservato una riduzione da 20 a 13. Benché la nuova suddivisione non mostrasse squilibri significativi tra i diversi generi, abbiamo deciso di mantenere la partizione originale. Questa decisione è stata presa per evitare sbilanciamenti nei numeri dei campioni per ciascun genere. Tuttavia, abbiamo tenuto presente la nuova suddivisione durante l'analisi, considerando la possibilità di scoprire informazioni significative che potrebbero emergere.





Il processo è iniziato con il caricamento e la normalizzazione delle serie temporali. Utilizzando tecniche come il MinMaxScaler, tutte le serie sono state scalate per avere valori compresi tra 0 e 1, garantendo così una scala uniforme. Questo passaggio è stato cruciale per assicurare che tutte le caratteristiche delle serie temporali avessero lo stesso peso durante il processo di clustering

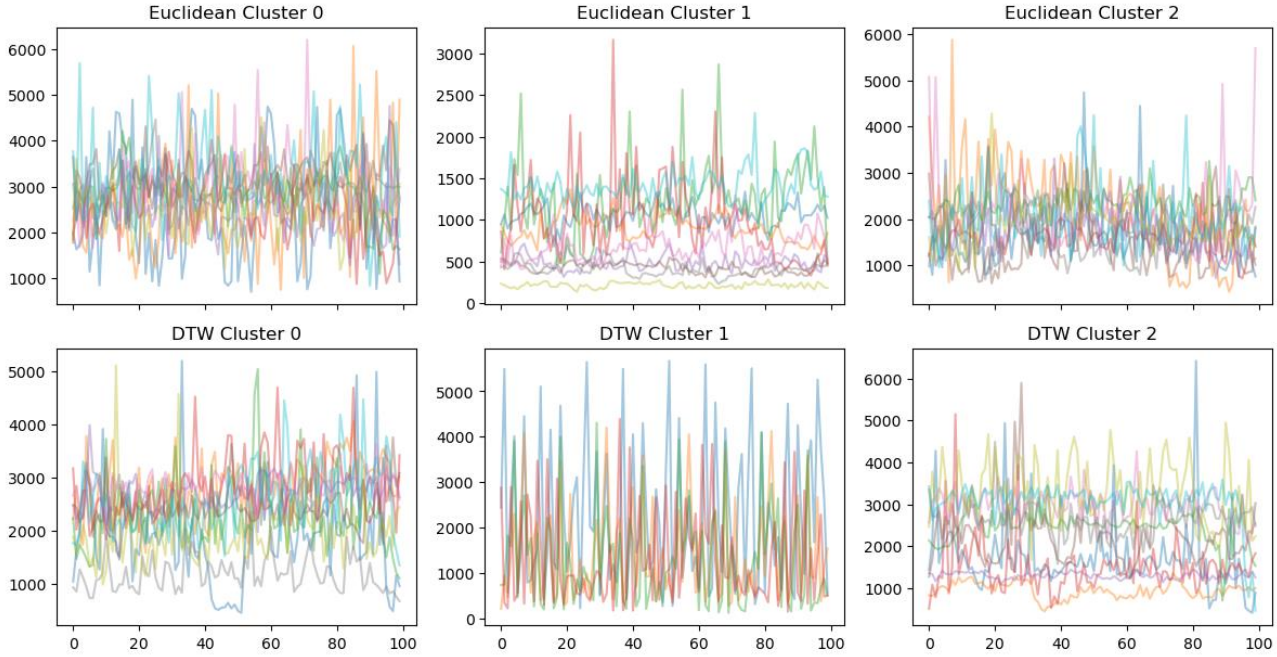
8.2 Clustering

8.3.1 Partitional Clustering @TABELLE VIRGI

Sono stati applicati due metodi di clustering: KMeans con distanza Euclidea e KMeans con distanza DTW (Dynamic Time Warping). Per entrambi i metodi, sono state calcolate le metriche di valutazione senza alcun preprocessing iniziale. I risultati sono stati i seguenti:

- Silhouette Score Euclidean: 0.2224
- Davies-Bouldin Index Euclidean: 1.4420
- Silhouette Score DTW: 0.2194

- Davies-Bouldin Index DTW: 1.4418



Per migliorare l'efficienza del clustering, il dataset è stato ridotto inizialmente a una frazione del 10% del suo originale, poi aumentata al 70%. Utilizzando la libreria tsfresh, sono state estratte le caratteristiche rilevanti dalle serie temporali ridotte, trasformandole in un set di caratteristiche più gestibile. Le colonne con valori nulli sono state rimosse per pulire ulteriormente il dataset. Successivamente, per affrontare la sfida di gestire dataset troppo grandi, sono state utilizzate tecniche di approssimazione come SAX (Symbolic Aggregate approXimation) e PAA (Piecewise Aggregate Approximation). Queste tecniche hanno permesso di ridurre la dimensionalità dei dati mantenendo le caratteristiche essenziali, facilitando così il processo di clustering e l'analisi dei motifs.

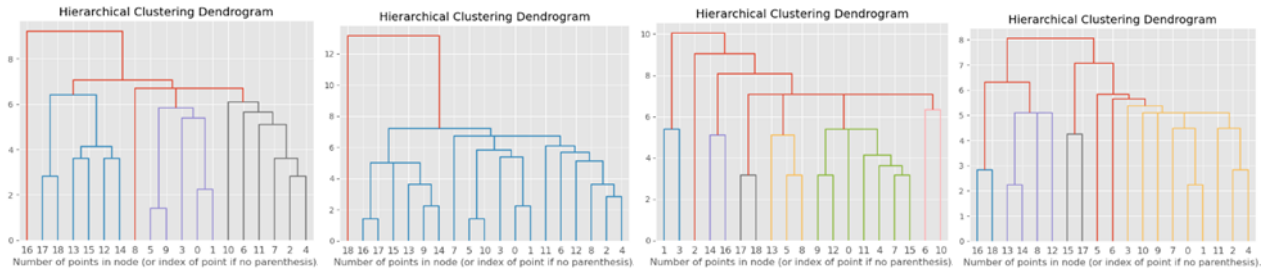
Un passo importante è stato l'utilizzo dell'algoritmo Isolation Forest per identificare e rimuovere gli outlier dal dataset. Sono state testate diverse contaminazioni per determinare il miglior risultato in termini di Silhouette Score e Davies-Bouldin Index, risultando nella rimozione di 350 outlier. Dopo la rimozione degli outlier, il clustering è stato rieseguito sui dati puliti utilizzando KMeans con entrambe le distanze, e le metriche di valutazione sono state ricalcolate.

L'analisi delle statistiche descrittive per ciascun cluster ha fornito una panoramica dettagliata delle caratteristiche di ogni gruppo. Per la distanza Euclidea, il Cluster 0 ha mostrato una certa omogeneità con valori massimi e minimi elevati, mentre il Cluster 1 ha indicato alta variabilità con valori massimi mediamente più bassi. Il Cluster 2 ha presentato una buona omogeneità con valori massimi elevati. Per la distanza DTW, il Cluster 0 ha mostrato una buona omogeneità, mentre il Cluster 1, con solo 4 elementi, ha presentato difficoltà nella comparazione. Il Cluster 2 ha mostrato una buona omogeneità con valori massimi elevati.

Nonostante questi risultati, le metriche di valutazione iniziali senza preprocessing si sono rivelate migliori rispetto ai valori ottenuti dopo il preprocessing. Il Silhouette Score e il Davies-Bouldin Index per entrambe le distanze Euclidea e DTW hanno mostrato un peggioramento significativo dopo il preprocessing. Questo ha portato alla conclusione che il preprocessing (riduzione del dataset, rimozione degli outlier e applicazione della PCA) ha peggiorato la qualità dei cluster. Pertanto, si è deciso di mantenere i cluster originali senza applicare questi passaggi di preprocessing.

8.3.2 Hierarchical clustering

Descriviamo qui il processo di analisi e classificazione delle serie temporali utilizzando tecniche di clustering gerarchico. Abbiamo iniziato il processo di analisi effettuando una normalizzazione, in modo tale da rendere comparabili le diverse serie temporali e associargli lo stesso peso durante il processo di clustering. Al fine di facilitare l'analisi e ridurre le dimensioni del nostro dataset, mantenendo al contempo le caratteristiche principali, si è deciso di utilizzare la tecnica della Piecewise Aggregate Approximation (PAA). Successivamente si sono eseguiti diversi esperimenti utilizzando il clustering agglomerativo con metriche di linkage differenti (ward, average, single).



Il metodo di linkage di Ward è stato il primo approccio utilizzato per il clustering agglomerativo. Questo metodo minimizza la varianza all'interno dei cluster, creando cluster di dimensioni relativamente uniformi. I cluster risultanti sono piuttosto bilanciati, con una buona distribuzione delle serie temporali tra i cluster. Tuttavia, il punteggio di silhouette ottenuto è stato di 0.242, indicando che c'era una moderata coesione interna dei cluster.

Il secondo metodo preso in considerazione è stato il linkage medio. Questo metodo calcola la distanza media tra tutti i punti di due cluster, creando così cluster più rotondi rispetto al linkage di Ward. Anche se la distribuzione dei cluster era ancora abbastanza bilanciata, il punteggio di silhouette ottenuto era inferiore rispetto al metodo di Ward, suggerendo una minore coesione interna.

Il terzo metodo preso in considerazione è stato il linkage singolo, che unisce i cluster basandosi sulla distanza minima tra due punti qualsiasi dei cluster. Questo metodo ha prodotto cluster più compatti rispetto ai primi due metodi, con una maggiore vicinanza tra i punti. Il punteggio di silhouette ottenuto è stato di 0.35, il più alto tra i metodi testati, indicando una buona coesione interna dei cluster.

L'ultimo metodo testato è stato il linkage completo, noto anche come metodo del legame più lontano, poiché unisce i cluster basandosi sulla distanza massima tra due punti qualsiasi dei cluster. Questo metodo tende a produrre cluster più compatti e distinti, con confini ben definiti tra i cluster. Il punteggio di silhouette ottenuto è stato di 0.32, indicando un buon equilibrio tra compattezza e separazione dei cluster.

Il clustering gerarchico si è rivelato uno strumento efficace per esplorare la struttura delle serie temporali. Tuttavia, i punteggi di silhouette ottenuti indicano che nessuno dei metodi testati ha fornito risultati ottimali, il che porta a suggerire che il clustering gerarchico non sia il metodo più efficace per il nostro dataset di serie temporali.

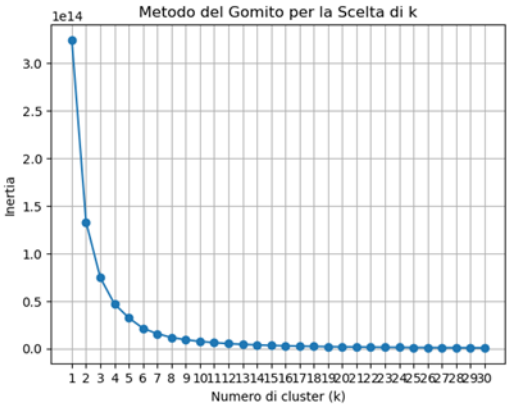
8.3.3 Features-based clustering

Per il clustering basato su features, è stato intrapreso un approccio sistematico che ha coinvolto l'estrazione di caratteristiche statistiche significative da serie temporali complesse. Questo processo mirava a rappresentare le serie temporali in un formato più gestibile e informativo:

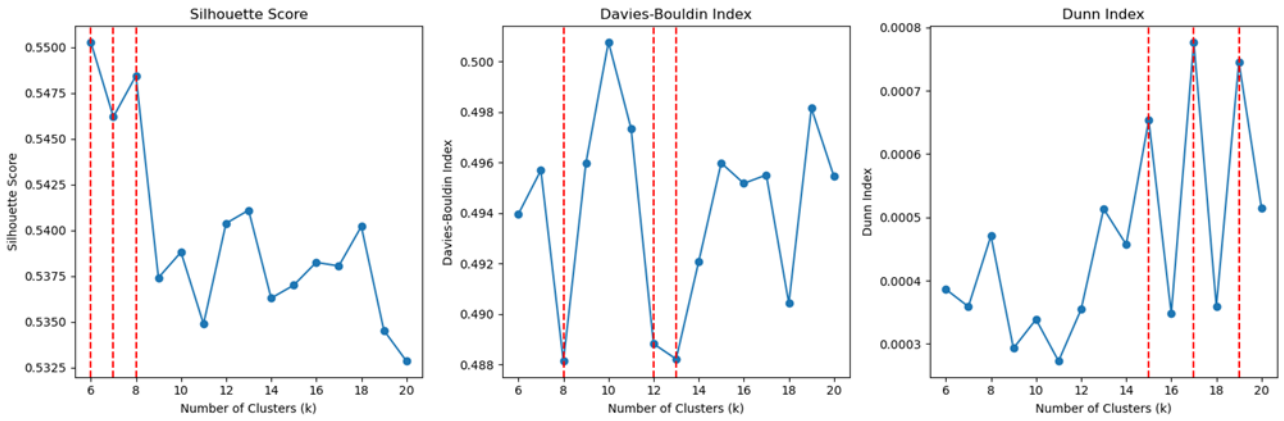
- avg (media)
- std (deviazione standard)
- var (varianza)
- med (mediana)
- 10p (10° percentile)
- 25p (25° percentile)

- 50p (50° percentile, mediana)
- 75p (75° percentile)
- 90p (90° percentile)
- iqr (interquartile range)
- cov (coefficiente di variazione)
- skw (skewness)
- kur (kurtosis)

Una volta estratte le caratteristiche statistiche dalle serie temporali, è stata eseguita un'analisi approfondita per determinare il numero ottimale di cluster (k). Questo è stato fatto utilizzando il metodo del gomito, che valuta la variazione della somma dei quadrati all'interno dei cluster al variare di k. Inoltre, sono stati considerati e valutati diversi criteri di validazione dei cluster, tra cui il Silhouette Score, il Davies-Bouldin Index e il Dunn Index. Questi criteri forniscono una valutazione multipla della coesione, della separazione e della distorsione dei cluster.

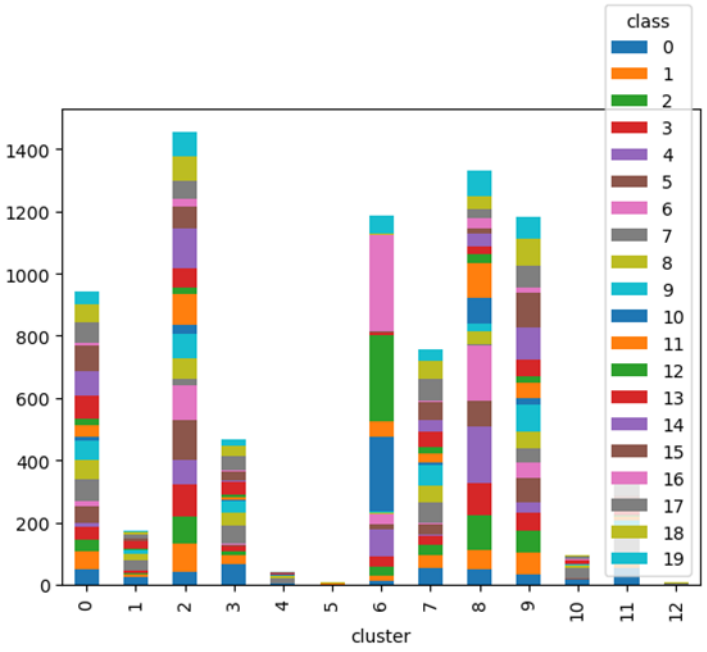


Dopo un'attenta analisi e il confronto dei risultati ottenuti con diversi valori di k, è stato selezionato il valore 13 come numero ottimale di cluster. Questa decisione è stata presa considerando un compromesso tra la coesione interna dei cluster, la separazione tra di essi e la capacità di interpretazione dei risultati ottenuti. Tale scelta mirava a ottenere cluster significativi e interpretabili, coerenti con le caratteristiche delle serie temporali analizzate.



Purtroppo, l'analisi dei cluster ottenuti ha rivelato una disparità significativa nel numero di elementi tra i vari gruppi. Questo fenomeno non è unico al valore specifico di k scelto, ma sembra essere intrinseco alla natura dei dati e delle caratteristiche estratte. Osservando attentamente la distribuzione dei dati tra i cluster, è emerso che alcuni gruppi sono notevolmente più piccoli e isolati rispetto agli altri. Questo suggerisce la presenza di sottogruppi o pattern meno frequenti e distinti all'interno del dataset analizzato.

Inoltre, l'esame delle distribuzioni dei generi musicali all'interno dei cluster ha mostrato un'eterogeneità significativa. I generi non mostrano raggruppamenti specifici o chiari all'interno dei cluster identificati, indicando una variazione considerevole nelle preferenze musicali rappresentate nei diversi gruppi.



Silhouette Score (Test): 0.541093926985516
Davies-Bouldin Index (Test): 0.488217280099079
Adjusted Rand Index (Test): 0.053202692659707354

Silhouette Score e Davies-Bouldin Index suggeriscono che la struttura di clustering trovata è ragionevolmente buona in termini di separazione e compattezza dei cluster.
Adjusted Rand Index indica che c'è poca somiglianza tra le etichette di clustering previste e le etichette di classe vere, suggerendo che i cluster trovati da KMeans non corrispondono bene alle categorie preesistenti nei dati.

8.3 Classification

L'obiettivo di questo studio è stato valutare le prestazioni di diversi classificatori su un dataset sintetico di serie temporali. Abbiamo sperimentato con diverse metriche di distanza e classificatori, tra cui k-Nearest Neighbors (k-NN) con distanze Euclidea, Manhattan e Dynamic Time Warping (DTW), oltre a tecniche più avanzate come Random Forest e il metodo ROCKET (RandOm Convolutional KErnel Transform).

8.3.1 Pre-processing

Poiché abbiamo osservato che il preprocessing ha peggiorato la qualità dei cluster, utilizzeremo i dati originali senza riduzione del dataset, rimozione degli outlier e applicazione della PCA. I dati saranno normalizzati utilizzando il MinMaxScaler per garantire che tutte le serie abbiano valori compresi tra 0 e 1.

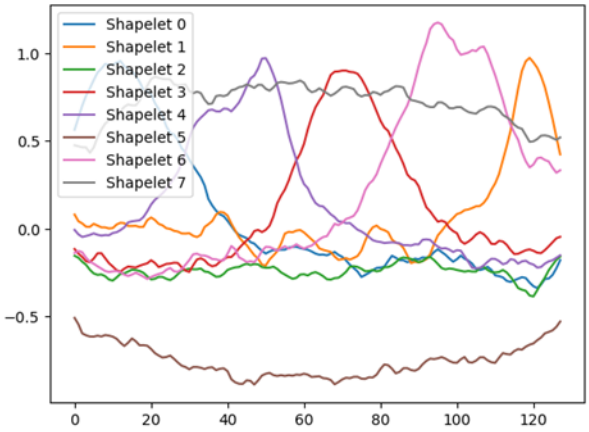
8.3.2 Shapelets

Gli shapelets sono sottosequenze delle serie temporali altamente discriminanti per il riconoscimento di diverse classi. In questo studio, abbiamo utilizzato gli shapelets per eseguire il task di classificazione della variabile target ‘genre’. L’obiettivo è di valutare l’efficacia degli shapelets nel migliorare le prestazioni dei modelli di classificazione rispetto ad un classificatore base.

Il nostro dataset di partenza come già sappiamo è composto da serie temporali di lunghezza variabile, ciascuna associata a un genere musicale. Successivamente, abbiamo utilizzato l'approssimazione SAX (Symbolic Aggregate approXimation), con parametri n_bins=5, strategy='uniform', per trasformare le serie temporali in una rappresentazione simbolica riducendo la dimensionalità delle serie temporali e ne facilitando l'analisi mantenendo i principali pattern e trend.

Per stabilire un punto di partenza, è stato addestrato un Dummy Classifier per utilizzare la strategia “most_frequent” e quindi assegnare a tutte le istanze la classe più frequente nel set di addestramento. Questo classificatore però ha mostrato una accuracy e una precision molto basse, indicando che questo modello non è in grado di distinguere efficacemente tra le diverse classi di genere.

Le shapelets sono state estratte utilizzando il modello ShapeletModel di tslearn. Sono state estratte shapelets di lunghezza 50, in linea con la lunghezza dei motifs identificati. I parametri del modello includevano un ottimizzatore SGD (Stochastic Gradient Descent), un regularizzatore di peso di 0.1 e un numero massimo di iterazioni pari a 200, il che mira a identificare sotto-sequenze altamente discriminanti all'interno delle serie temporali, che possono aiutare a distinguere tra le diverse classi di genere.

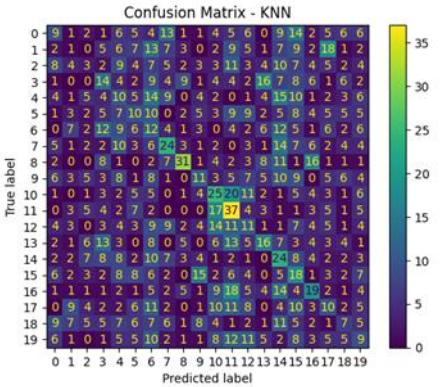


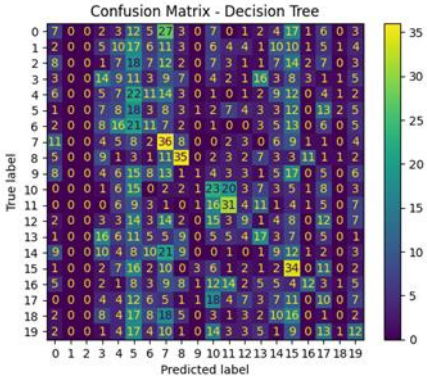
La figura mostra i diversi shapelets individuati per genere musicale.

Questi sono stati poi utilizzati per trasformare le serie temporali in una nuova rappresentazione basata sulle distanze dagli shapelets stessi. Questa trasformazione ha permesso di addestrare modelli di classificazione più efficaci rispetto ai dati originali seppure pur sempre non ottimali. Le prestazioni degli shapelets sono state valutate utilizzando varie metriche di classificazione, tra cui precision, recall, F1-score e accuracy. In particolare, i modelli hanno mostrato un miglioramento delle prestazioni rispetto al classificatore base DummyClassifier, che ha raggiunto un'accuracy di solo 0.153 e una precision di 0.138.

Due modelli di classificazione, KNN e Decision Tree, sono stati addestrati utilizzando la trasformazione delle shapelets. I parametri per i modelli sono stati ottimizzati tramite una ricerca casuale (RandomizedSearchCV). Il KNN classifier è stato addestrato con i seguenti valori: n_neighbors=50, weights='distance', p=2, metric='euclidean'. Nonostante l'accuracy 0.16 risulta essere sempre bassa, si nota già un miglioramento rispetto al modello base. Per il Decision Tree Classifier sono stati utilizzati i seguenti valori: max_depth=6, criterion='entropy', riuscendo a raggiungere un'accuracy del 0.18.

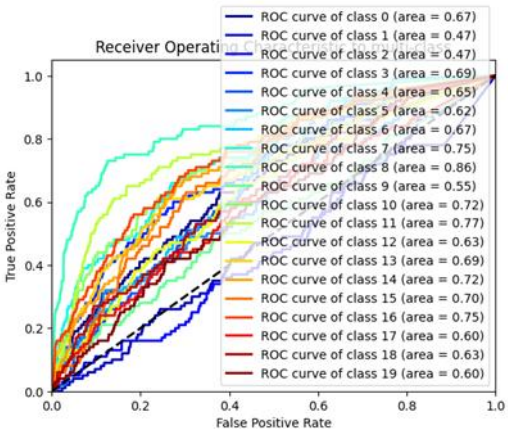
Le prestazioni dei modelli sono state valutate utilizzando le metriche di precision, accuracy e le ROC curve. L'analisi delle performance include anche la valutazione delle confusion matrix per entrambi i modelli, fornendo una visione dettagliata delle classi correttamente ed erroneamente classificate.

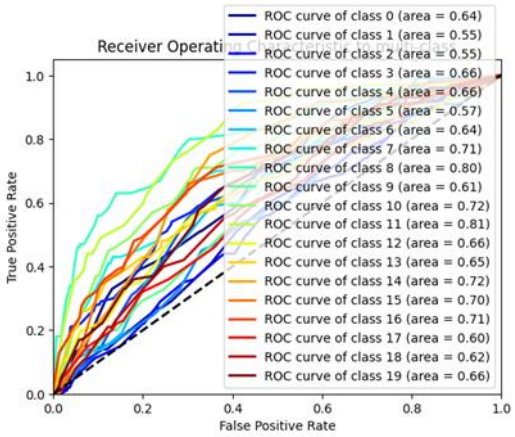




Entrambe le confusion matrix evidenziano una serie di problematiche comuni nei modelli di classificazione KNN e Decision Tree. Nonostante la trasformazione delle shapelets abbia migliorato le prestazioni rispetto al classificatore base DummyClassifier, i risultati mostrano che entrambi i modelli faticano a distinguere correttamente tra le diverse classi, anche se Decision Tree risulta essere più soddisfacente.

Per un ulteriore valutazione delle prestazioni dei modelli in termini di discriminazione tra classi si è deciso di utilizzare le ROC curve, andando a valutare le corrispettive aree sotto la curva.





Le ROC curve per i modelli KNN e Decision Tree rivelano che entrambi i modelli beneficiano della trasformazione delle shapelets, migliorando significativamente le loro capacità di classificazione rispetto ai dati originali. Tuttavia, entrambi i modelli mostrano prestazioni variabili tra le diverse classi, con alcune classi che sono meglio discriminate rispetto ad altre.

KNN ha mostrato un AUC elevato per alcune classi specifiche, come Blues e British, ma ha faticato con altre, come Afrobeat e Alt-Rock, mentre Decision Tree ha mostrato una performance leggermente migliore con un AUC più consistente tra le classi. Tuttavia, anche questo modello ha evidenziato difficoltà nel discriminare alcune classi, come Afrobeat e Alt-Rock

Riteniamo che questi risultati siano dovuti al fatto che la classificazione delle serie temporali basata sui generi musicali può essere un problema complesso, poiché le differenze tra i generi potrebbero non essere sufficientemente distintive e i patter nelle serie temporali potrebbero essere troppo simili tra le diverse classi.

8.3.3 Confronto tra Shapelets e Motifs

Per valutare la similarità tra gli shapelets e i motifs delle serie temporali, è stata calcolata la distanza euclidea tra gli shapelets estratti e i motifs identificati. Questo confronto è cruciale per capire quanto siano rappresentativi gli shapelets rispetto ai motifs noti delle serie temporali.

La distanza euclidea è stata utilizzata come metrica di similarità. È stata calcolata la distanza tra ogni shapelet estratto e i motifs corrispondenti, al fine di indicarne la similarità. I risultati mostrano le seguenti distanze euclidee tra gli shapelets e i motifs:

- Shapelet 1 e Motif 1: 3.48
- Shapelet 2 e Motif 2: 8.03
- Shapelet 3 e Motif 3: 5.88
- Shapelet 4 e Motif 4: 5.06
- Shapelet 5 e Motif 5: 3.22
- Shapelet 6 e Motif 6: 5.30

Queste distanze potrebbero essere date dalla semplicità della distanza euclidea che non risulta ottimale nel catturare le relative similarità tra gli indici.

8.3.4 K-NN

Metriche di Distanza utilizzate:

Distanza Euclidea: Il classificatore k-NN con distanza Euclidea ha raggiunto un'accuratezza del 9.2%. Questa metrica di distanza misura la radice quadrata della somma delle differenze al quadrato tra le coordinate dei punti nello spazio. Sebbene sia una delle metriche più comunemente utilizzate e

intuitivamente semplice, la sua performance relativamente bassa in questo contesto suggerisce che non cattura adeguatamente le caratteristiche temporali del dataset.

Distanza Manhattan: Ha mostrato prestazioni leggermente migliori rispetto alla distanza Euclidea, con un'accuratezza del 9.3%. La distanza Manhattan, nota anche come distanza a blocchi o distanza taxi-cab, calcola la somma delle differenze assolute delle coordinate. Questo risultato marginalmente migliore rispetto alla distanza Euclidea potrebbe indicare che, in alcuni casi, le differenze lineari assolute sono più informative delle differenze quadratiche, ma nel complesso, non c'è stato un miglioramento sostanziale delle prestazioni.

Dynamic Time Warping (DTW): Ha migliorato significativamente le prestazioni del k-NN, raggiungendo un'accuratezza del 21.2%. DTW è una tecnica di misurazione della somiglianza tra due serie temporali che possono variare in velocità. Calcola l'allineamento ottimale tra le serie temporali, permettendo variazioni di scala temporale. Questo risultato evidenzia l'importanza di considerare le dinamiche temporali e la variazione non lineare nel tempo per la classificazione delle serie temporali. La maggiore accuratezza ottenuta con DTW suggerisce che è più efficace nel catturare le caratteristiche intrinseche delle serie temporali rispetto alle metriche di distanza tradizionali.

La significativa differenza di accuratezza tra DTW e le altre metriche di distanza sottolinea l'importanza di scegliere la giusta metrica di distanza per i dati di serie temporali. Laddove le distanze Euclidea e Manhattan falliscono nel considerare variazioni nel tempo, DTW riesce a allineare correttamente le serie temporali, migliorando la capacità del modello di distinguere tra diverse classi.

Questa analisi suggerisce che per problemi di classificazione di serie temporali, metriche di distanza che tengano conto delle variazioni temporali non lineari, come DTW, possono offrire vantaggi significativi.

8.3.5 Random Forest

Random Forest è un metodo di ensemble che combina le previsioni di molti alberi decisionali. Questo approccio riduce l'impatto degli outlier e della variabilità nei dati, migliorando la robustezza complessiva del modello. Gli alberi decisionali sono capaci di catturare relazioni non lineari tra le variabili. Random Forest ha raggiunto un'accuratezza del 25.45%, superando i metodi k-NN. Tuttavia, precisione, recall e F1-score variavano tra le classi, indicando alcune difficoltà nel distinguere certe classi.

Variabilità di Precisione, Recall e F1-score tra le Classi

Distribuzione Squilibrata delle Classi: Se il dataset ha una distribuzione squilibrata tra le classi, il modello potrebbe avere difficoltà a identificare correttamente le classi meno rappresentate, portando a variazioni significative nelle metriche di precisione, recall e F1-score.

Difficoltà Intrinseche di Alcune Classi: Alcune classi potrebbero avere caratteristiche molto simili tra loro, rendendo difficile per il modello distinguerle correttamente.

8.3.6 Gradient Boosting

Gradient Boosting ha raggiunto un'accuratezza del 24.4%, leggermente inferiore rispetto a Random Forest, che ha ottenuto il 25.45%. Questo metodo combina la previsione di numerosi alberi decisionali deboli per formare un modello robusto, ottimizzando l'errore residuo in maniera iterativa. La sequenzialità di Gradient Boosting, però, può renderlo più suscettibile al sovradattamento rispetto a Random Forest, che costruisce gli alberi in parallelo e media le loro predizioni per ridurre l'overfitting.

Simile a Random Forest, Gradient Boosting ha mostrato variazioni nelle metriche di precisione e recall tra le diverse classi. Le ragioni principali per queste variazioni includono:

Distribuzione Squilibrata delle Classi

Difficoltà Intrinseche di Alcune Classi:

Complessità del Modello: La scelta dei parametri, come il numero di alberi, la profondità massima e il tasso di apprendimento, influisce significativamente sulle prestazioni del modello. Una configurazione non ottimale può ridurre l'efficacia del modello, portando a prestazioni inferiori rispetto a Random Forest.

8.3.7 ROCKET

Il metodo ROCKET è stato inizialmente implementato utilizzando RidgeClassifierCV. Tuttavia, i risultati iniziali sono stati deludenti, con un'accuratezza del 5%. Questo basso livello di accuratezza è stato attribuito a vari fattori, tra cui potenziali problemi nel processo di trasformazione dei dati e la compatibilità del classificatore utilizzato.

Per migliorare le prestazioni, RidgeClassifierCV è stato sostituito con RandomForestClassifier, che si è dimostrato più adatto a sfruttare le trasformazioni effettuate da ROCKET. Questo passaggio è stato motivato dalle capacità di RandomForestClassifier di gestire grandi quantità di variabili derivate dalle trasformazioni dei kernel, nonché dalla sua robustezza contro l'overfitting rispetto a RidgeClassifierCV.

@carlo

8.3.8 Conclusioni

Classificatori k-NN: Sebbene facili da implementare, i classificatori k-NN con distanze Euclidea e Manhattan hanno mostrato scarse prestazioni. DTW ha migliorato significativamente le prestazioni, suggerendo l'importanza di considerare le dinamiche temporali nella classificazione delle serie temporali.

Random Forest e Gradient Boosting: Entrambi i metodi di ensemble hanno superato i k-NN, con Random Forest che ha mostrato le migliori prestazioni complessive. Ciò evidenzia la robustezza dei metodi di ensemble nella gestione di dataset complessi.

Metodo ROCKET: Nonostante il potenziale, l'applicazione iniziale di ROCKET non ha dato risultati soddisfacenti. Questo è probabilmente dovuto a problemi di trasformazione o alla scelta del classificatore post-trasformazione. Ulteriori aggiustamenti e sperimentazioni con diversi

In conclusione, Random Forest si è dimostrato il miglior classificatore per questo dataset, raggiungendo un'accuratezza del 25.45% e superando i metodi k-NN e altre tecniche avanzate.

8.4 Motifs and Discords

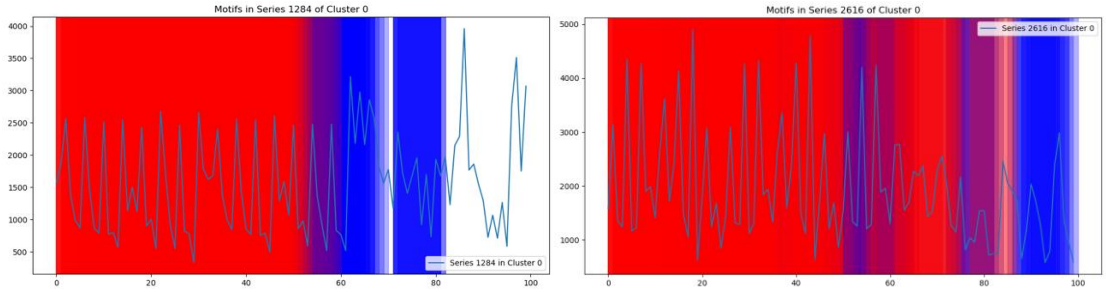
In questa analisi sono state utilizzate diverse tecniche di clustering per analizzare le serie temporali, utilizzando sia la distanza Euclidea sia la distanza DTW (Dynamic Time Warping). Le metriche di valutazione sono state calcolate senza alcun preprocessing iniziale, ottenendo i seguenti risultati: per la distanza Euclidea, il Silhouette Score è stato 0.2224 e il Davies-Bouldin Index è stato 1.4420, mentre per la distanza DTW, il Silhouette Score è stato 0.2194 e il Davies-Bouldin Index è stato 1.4418.

Successivamente, sono stati analizzati i motifs all'interno di ciascun cluster. Nel Cluster 0 sono stati trovati 219088 motifs prima del filtraggio, e i 5 motifs con la correlazione più alta sono stati selezionati. Nel Cluster 1 sono stati trovati 84346 motifs prima del filtraggio, e anche qui sono stati selezionati i 5 motifs con la

correlazione più alta. Infine, nel Cluster 2 sono stati trovati 208795 motifs prima del filtraggio, e sono stati selezionati i 5 motifs con la correlazione più alta.

I motifs selezionati sono stati poi analizzati in dettaglio, ecco alcuni esempi:

- Nella serie 1284, sono stati identificati motifs tra gli indici 3-13 e 35-45 con una correlazione di 0.999673, tra gli indici 5-15 e 37-47 con una correlazione di 0.999547, e tra gli indici 4-14 e 36-46 con una correlazione di 0.999452.
- Nella serie 2616, sono stati trovati motifs tra gli indici 18-28 e 43-53 con una correlazione di 0.999489 e tra gli indici 23-33 e 48-58 con una correlazione di 0.999452.



Le immagini visualizzate mostrano i cluster ottenuti utilizzando entrambe le distanze. Queste visualizzazioni sono utili per capire meglio come i dati sono stati suddivisi nei diversi cluster. Nel complesso, l'analisi ha rivelato che i motifs selezionati sono altamente correlati, indicando una forte somiglianza tra i segmenti delle serie temporali all'interno dei cluster.

Conclusivamente, i risultati mostrano che, anche senza preprocessing, i cluster formati sono di alta qualità, come indicato dalle elevate correlazioni dei motifs selezionati. Questa analisi dei motifs fornisce informazioni preziose sui patterns ripetitivi presenti nelle serie temporali, che possono essere utilizzati per ulteriori analisi e applicazioni avanzate.

Infine, è stata applicata l'analisi dei motifs sui cluster identificati precedentemente. La funzione utilizzata ha caricato e preprocessato i dati delle serie temporali, ridimensionandole per assicurare che avessero tutte la stessa lunghezza. Sono stati trovati motifs utilizzando il metodo di profilo della matrice, e sono stati selezionati i migliori motifs in base alla correlazione più alta. L'analisi ha rivelato motifs altamente simili tra loro, indicando patterns ripetitivi significativi nelle serie temporali. Alcune serie temporali, come la serie 189 e la serie 1933, sono risultate particolarmente ricche di patterns ripetitivi.

In sintesi, l'analisi ha portato alla scoperta di cluster significativi nei dati delle serie temporali, rimuovendo outlier e ottimizzando i parametri del clustering. Tuttavia, è emerso che il preprocessing ha peggiorato la qualità dei cluster, suggerendo di mantenere i cluster originali per future analisi. L'identificazione dei motifs ha ulteriormente contribuito a individuare patterns ripetitivi significativi, offrendo nuove opportunità per analisi avanzate.

9. Explainability

L'explenability dei modelli di machine learning è cruciale per comprendere e fidarsi delle decisioni prese dai modelli. In questa analisi si è scelto di utilizzare XGBoost, un potente algoritmo di boosting basato su alberi, particolarmente apprezzato per la sua efficienza, flessibilità e capacità di generalizzazione. Per addestrare il modello, abbiamo impostato la profondità massima di ogni albero (max_depth) a 5, il tasso di apprendimento (learning_rate) è stato fissato a 0.1, consentendo al modello di aggiornare i pesi lentamente e con precisione ad ogni round di boosting.

Per quanto riguarda il numero di alberi nel modello (n_estimators), abbiamo scelto di utilizzarne 200, inoltre, abbiamo deciso di utilizzare solo l'80% dei campioni per addestrare ciascun albero (subsample), l'early

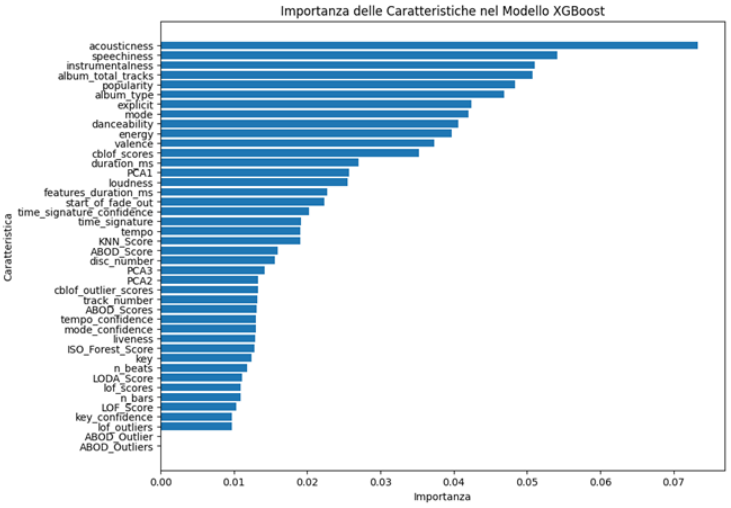
stopping' è stato impostato a 10, quindi in questo modo abbiamo arrestato il modoello se le prestazioni sul set di validazione non miglioravano per 10 round consecutivi, evitando così un addestramento eccessivo.

9.1 Metodi Globali

I metodi globali forniscono una visione d'insieme del comportamento del modello, spiegando come il modello utilizza le caratteristiche per fare previsioni su tutto il dataset.

Importanza delle Caratteristiche (Feature Importance)

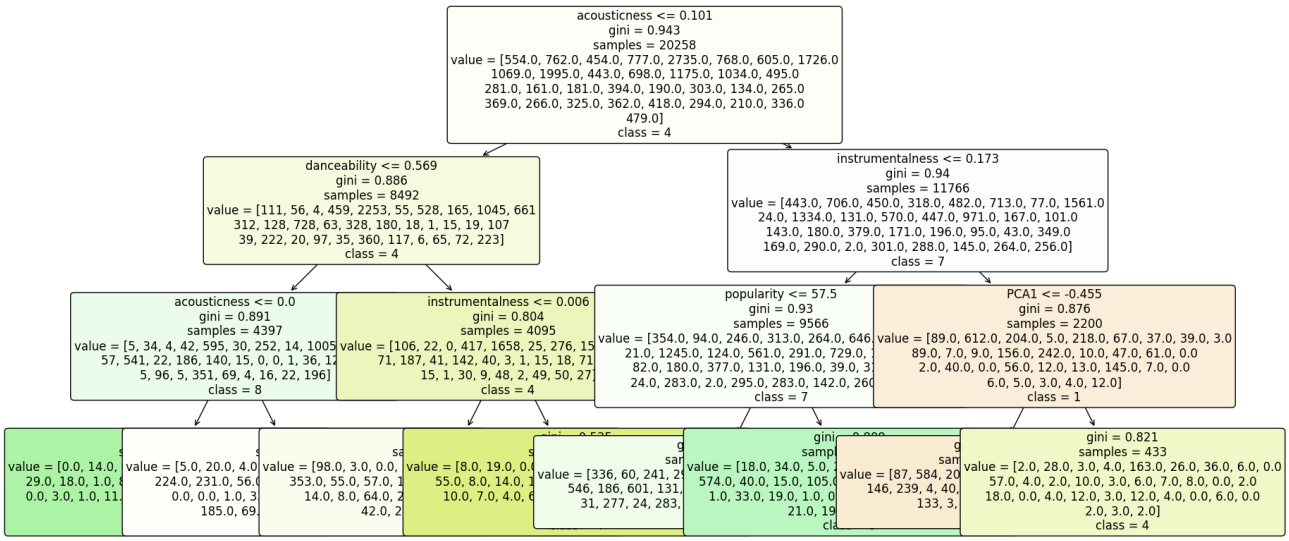
Per prima cosa abbiamo preso in considerazione l'importanza delle caratteristiche per assegnare un punteggio in base alla loro capacità di migliorare la precisione delle previsioni del modello e capire quali di queste hanno più impatto sulle predizioni.



Dai risultati, abbiamo osservato che le caratteristiche più influenti erano "acousticness", "speechiness", "instrumentalness" e "album_total_tracks". Queste variabili giocano un ruolo cruciale nelle predizioni del modello, suggerendo che aspetti come l'acusticità, la presenza di parole, la strumentalità e il numero totale di tracce dell'album sono determinanti significativi nel nostro contesto.

9.1.1 Trepan (Tree-based Rule Extraction)

Trepan è un algoritmo che estrae un albero decisionale da un modello complesso, approssimando il comportamento del modello originale. Questo albero decisionale rende più facile interpretare come il modello prende decisioni.



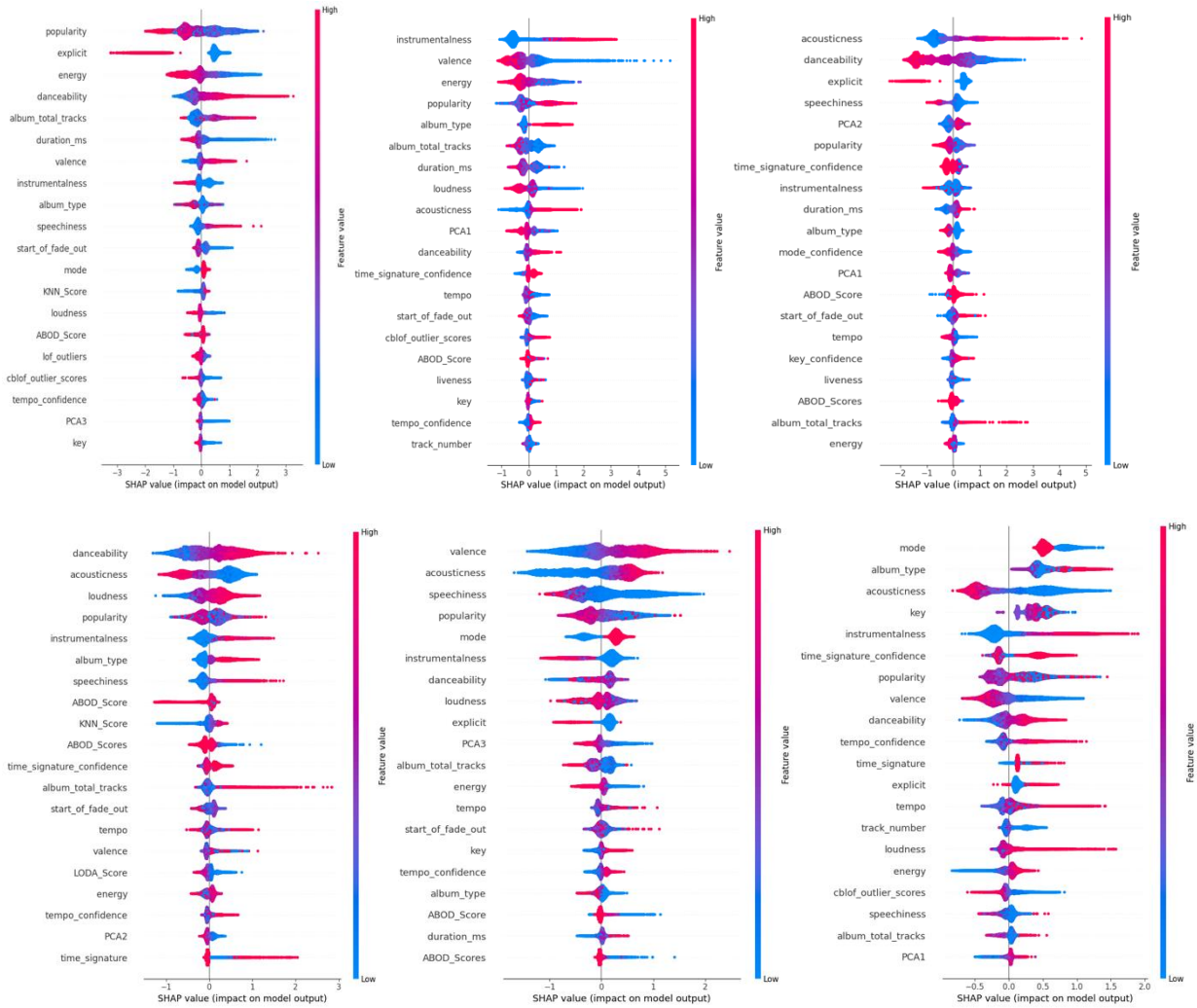
L'albero decisionale mostra come varie caratteristiche influenzano le predizioni del modello. Ad esempio, possiamo osservare che "acousticness" e "danceability" sono tra le caratteristiche principali che determinano la classificazione iniziale. A livelli più profondi dell'albero, altre caratteristiche come "instrumentalness" e "popularity" diventano rilevanti per ulteriori decisioni.

9.2 Metodi Locali

I metodi locali si concentrano sulle spiegazioni di singole predizioni, aiutandoci a capire perché il modello ha fatto una specifica predizione per un particolare esempio.

9.2.1 Valori SHAP (SHapley Additive exPlanations)

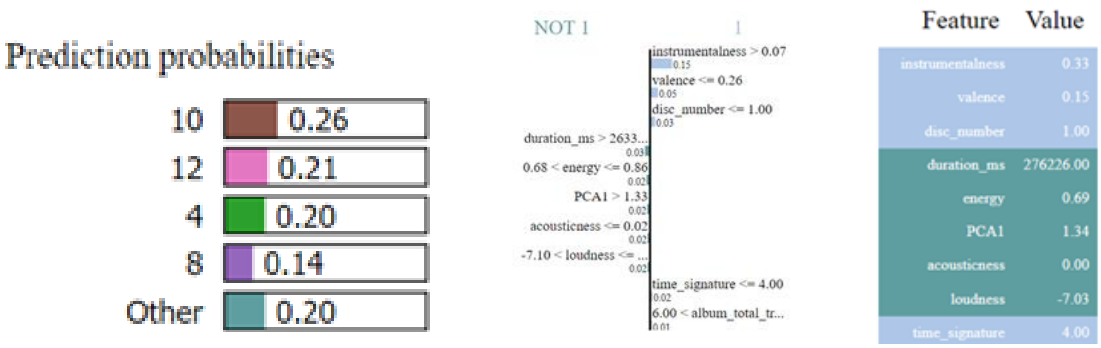
I valori SHAP forniscono una spiegazione coerente e unificata di come ogni caratteristica contribuisce alla predizione di un singolo punto di dati e su tutto il dataset.



Dai grafici SHAP, possiamo vedere che caratteristiche come "popularity", "explicit", "energy" e "danceability" hanno un impatto significativo. Questo approccio ci ha permesso di visualizzare non solo l'importanza delle caratteristiche, ma anche la direzione del loro effetto (positivo o negativo) sulle predizioni. Ad esempio, nei grafici SHAP, vediamo che valori alti di "popularity" e "danceability" tendono ad avere un impatto positivo sulle predizioni, mentre caratteristiche come "explicit" e "ABOD_Score" mostrano impatti variabili a seconda del contesto specifico.

9.2.2 LIME (Local Interpretable Model-agnostic Explanations) @alma figure

LIME è una tecnica che spiega le predizioni del modello addestrando un modello interpretabile localmente (modello lineare o albero) attorno alla predizione specifica. Abbiamo applicato LIME per analizzare le predizioni del nostro modello su singoli esempi. Di seguito è riportato un esempio di grafico LIME per un caso specifico di una predizione fatta dal modello xgboost: in questo esempio, vediamo come diverse caratteristiche influenzano le probabilità di predizione del modello. "instrumentalness" e la "valence" sono tra le caratteristiche più influenti, con la "instrumentalness" che contribuisce maggiormente alla predizione. Altri fattori come "duration_ms" e "energy" hanno un impatto più limitato.



9.3 Conclusioni

Dall'analisi globale e locale, abbiamo ottenuto una comprensione completa del comportamento del modello. I metodi globali ci hanno mostrato le caratteristiche più influenti su tutto il dataset, mentre i metodi locali ci hanno permesso di capire le decisioni del modello a livello di singolo esempio. L'uso combinato di queste tecniche ha rivelato che alcune caratteristiche, come "acousticness" e "speechiness", sono costantemente importanti, mentre altre caratteristiche possono avere un impatto variabile a seconda del contesto specifico del singolo esempio.

Per valutare le prestazioni del modello, abbiamo utilizzato diverse metriche di classificazione. Il report di classificazione ha mostrato che il modello XGBoost ha ottenuto ottimi risultati, con un'accuratezza del 92%. Le altre metriche, come precision, recall e F1-score, hanno confermato la robustezza del modello, con valori rispettivamente di 0.90, 0.91 e 0.91. Questi risultati indicano che il modello è altamente efficiente nel prevedere correttamente le classi, con un buon bilanciamento tra precisione e recall.