**DOCUMENTAZIONE CASO DI STUDIO**

**INGEGNERIA DELLA CONOSCENZA 2023-2024**

**Progetto realizzato da:**

**-Alessandro Olivieri, matricola: 703269,** [**a.olivieri14@studenti.uniba.it**](mailto:a.olivieri14@studenti.uniba.it)

**-Antonio Silvestre, matricola: 697697, a.silvestre2@studenti.uniba.it**

**Link allo zip del progetto:**

**Indice**

1. Introduzione
2. Dataset e Features
3. Modelli di classificazione
   * K-Nearest Neighbors
   * GaussianNB
   * Random Forest
   * Extra Trees
4. Ottimizzazione degli Iperparametri
   * Curva di validazione
   * Exhaustive grid search
5. Accuratezze dei Classificatori
   * Metriche di Valutazione
   * Risultati
6. Clustering
   * Algoritmo KMeans
   * Valutazione con Purity
7. Sviluppi futuri

**1. Introduzione**

L’obiettivo principale di questo progetto è sviluppare un sistema in grado di riconoscere il genere musicale di tracce audio in formato WAV. Il riconoscimento avviene tramite il confronto e la valutazione delle prestazioni di vari modelli di classificazione supervisionata. Tra i modelli esaminati vi sono: K-Nearest Neighbors (KNN), Naive Bayes (GaussianNB), Random Forest, e Extra Trees. Durante l’analisi, si è osservato che l’Extra Trees Classifier fornisce, nella maggior parte dei casi, le migliori performance in termini di accuratezza, precisione e richiamo.

L’implementazione si focalizza su un insieme diversificato di generi musicali come blues, classical, rock, hip hop, reggae, disco, country, metal, pop, e jazz. Questi generi vengono identificati sfruttando specifiche caratteristiche audio estratte direttamente dai file.

**2. Dataset e Features**

Il dataset utilizzato è disponibile su Kaggle, denominato “music features” e contiene 1000 tracce audio, suddivise in 800 per l’addestramento e 200 per il test. Ogni brano è descritto da una serie di feature ottenute tramite tecniche avanzate di estrazione, quali:

* **Tempo**: Indica la velocità con cui si sviluppa la musica, misurato in battiti per minuto.
* **Beats**: Rappresenta l'unità ritmica fondamentale della traccia.
* **Chroma\_stft**: Si basa sulla trasformata di Fourier a breve termine per identificare l'intensità delle diverse tonalità cromatiche.
* **RMSE**: Una misura dell'energia del segnale audio.
* **Spectral Centroid**: Indica il punto di equilibrio dello spettro di frequenze.
* **Spectral Bandwidth**: Definisce la gamma di frequenze attorno al centroide spettrale.
* **Roll-off**: Rappresenta l’attenuazione progressiva delle frequenze, cruciale per caratterizzare il timbro della musica.
* **Zero Crossing Rate**: Calcola la frequenza con cui il segnale attraversa lo zero, utile per determinare la tonalità della musica.
* **MFCCs**: I coefficienti Mel-frequency cepstral sono fondamentali per la rappresentazione del timbro sonoro e sono largamente impiegati in applicazioni di riconoscimento audio.

La fase di pre-elaborazione ha evidenziato che ogni caratteristica contribuisce in modo significativo alla classificazione, rendendo impraticabile la riduzione delle dimensioni del dataset.

Immagine che contiene Diagramma, linea, diagramma, schermata

Descrizione generata automaticamente

**3. Modelli di classificazione**

Per realizzare una classificazione precisa, sono stati utilizzati diversi modelli basati su algoritmi di apprendimento supervisionato, implementati con la libreria sklearn:

* *K-Nearest Neighbors*:

è un algoritmo utilizzato nel riconoscimento di pattern per la classificazione di oggetti basandosi sulle caratteristiche dei k oggetti più vicini a quello considerato. Un oggetto è classificato in base alla maggioranza dei voti dei suoi *k* vicini. *k* è un intero positivo tipicamente non molto grande. La scelta di k dipende dalle caratteristiche dei dati. Generalmente all'aumentare di k si riduce il rumore che compromette la classificazione. Al fine dell’apprendimento lo spazio multidimensionale viene partizionato in regioni in base alle posizioni e alle caratteristiche degli oggetti di apprendimento, rappresentati come vettori. Un oggetto è assegnato alla classe C se questa è la più frequente fra i k esempi più vicini all'oggetto sotto esame, la vicinanza si misura in base alla distanza fra punti. I vicini sono presi da un insieme di oggetti per cui è nota la classificazione corretta.

* *GaussianNB*:

I classificatori basati sul modello Naïve Bayes, utilizzano il teorema di Bayes:

Dove *P(y|x1,…,xn)* è la probabilità a posteriori, *P(y)* è la probabilità a priori, *P(x1,…,xn|y)* è la verosimiglianza e *P(x1,…,xn)* è la funzione di partizione. Nell’ utilizzo del classificatore GaussianNB, si presume che la probabilità delle feature sia gaussiana:

I parametri *σy* e *μy* sono stimati usando la massima probabilità.

* *Random Forest:*

È un modello d'insieme ottenuto dall'aggregazione tramite bagging di alberi di decisione. Esso è un meta-stimatore che si adatta ad una serie di alberi decisionali addestrati su vari sotto-campioni del dataset e utilizza la media di ogni singolo output di ogni albero per migliorare l’accuratezza predittiva e il controllo del sovradattamento. Il Random Forest deve essere dotato di due matrici: una matrice X sparsa che contiene i campioni di addestramento e una matrice Y di dimensioni che contiene i valori target.

* *Extra Trees:*

Tale modello è simile al precedente, la differenza risiede nella scelta degli alberi, la quale avviene in maniera puramente casuale.

**4. Ottimizzazione degli Iperparametri**

Si è eseguito un processo di ottimizzazione degli iperparametri con lo scopo di rendere i vari modelli di classificazione più accurati.  
Inizialmente in ogni modello di classificazione sono inclusi parametri con specifici valori, detti iperparametri.

Se questi ultimi non vengono espressi, ai parametri si assegneranno valori standard di default, che non aiutano il modello ad accrescere la propria accuratezza.  
Per stabilire gli iperparametri, si è fatto uso di diverse tecniche di ottimizzazione ovvero:  
  
***Curva di validazione*:**

metodo vantaggioso per evidenziare i valori teoricamente ottimizzati di ciascun modello.  
Essa può essere disegnata su un grafico, per mostrare il comportamento che il modello assume cambiando i valori di un singolo iperparametro.  
  
Immagine che contiene diagramma, Diagramma, linea

Descrizione generata automaticamente

Tale grafico, mostra come avviene la validazione per il parametro “max depth” presente all’interno del modello di classificazione “ExtraTrees Classifier”.  
Sull’asse delle ascisse vi è il parametro che si vuole settare in base ai valori che può avere, sulle ordinate vi è il valore di accuratezza.  
Le curve di training score e cross-validation score raffigurano il concetto fondamentale di questa procedura, infatti, in base a queste si può verificare per quale valore dell’iperparametro, diventa massima l’accuratezza.  
  
***Exhaustive grid search*:**    
 Questo metodo genera i possibili iperparametri attraverso una griglia di valori definita dal parametro “param\_grid”, formato da una gamma di valori per ogni parametro definito dall’utente.  
Vengono valutate le varie combinazioni di assegnazioni degli iperparametri in modo da scegliere quella migliore e mostrarla al termine del processo.

**5. Accuratezze dei Classificatori**

Ogni classificatore è stato valutato utilizzando metriche standard come accuratezza, precisione, richiamo, e F1-measure. I risultati sono stati raccolti e analizzati per determinare l'efficienza di ciascun modello.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Classificatori | Accuracy | Precision | Recall | F1-measure |
| *K-NN* | 0.365 | 0.371 | 0.365 | 0.356 |
| *GaussianNB* | 0.435 | 0.404 | 0.435 | 0.389 |
| *RandomForest* | 0.530 | 0.535 | 0.530 | 0.505 |
| *ExtraTrees* | 0.670 | 0.665 | 0.670 |  |

Il confronto tra i classificatori ha evidenziato la superiorità dell’Extra Trees Classifier, che ha mostrato le migliori capacità di predizione. La matrice di confusione e le metriche dettagliate confermano l'efficacia del modello nel distinguere i generi musicali.

1. **Clustering**

È stata effettuata una procedura di clustering basata sull’algoritmo *KMeans*. Tale algoritmo ha come scopo quello di minimizzare la varianza totale intra-gruppo, e nel nostro caso è stato utile per raggruppare, in cluster diversi, brani appartenenti a generi musicali diversi.

Prima di procedere, abbiamo portato sulla stessa scala le varie input features.

Una volta eseguito il clustering, è stato possibile valutare i risultati grazie alla metrica denominata *purity*. Per calcolare questa metrica, ogni cluster è assegnato alla classe che è più frequente nel cluster, e successivamente l’accuratezza di questa assegnazione è misurata contando il numero di tracce assegnate in modo corretto, e dividendo per il numero di esempi.

In particolare, la purity è rappresentata da un punteggio nell’intervallo [0,1], che indica in quale misura ognuno dei cluster contenga una ed una sola classe. Di conseguenza, ad un valore maggiore di tale metrica corrisponde una maggiore omogeneità dei cluster, e dunque un partizionamento in classi più accurato.

Immagine che contiene Carattere, bianco, testo, design

Descrizione generata automaticamente

1. **Sviluppi futuri**