**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: https://tinyurl.com/yc3yty2t**

**Indice**

1. **Introduzione**
2. **Linguaggio e Ambienti Utilizzati**
   * **Librerie Utilizzate**
3. **Manuale d'Uso**
   * **Preparazione del Dataset**
   * **Esecuzione del Programma**
   * **Analisi dei Risultati**
   * **Predizione su Nuovi Brani**
   * **Ripetizione delle Predizioni**
   * **Visualizzazione dell’Importanza delle Caratteristiche**
4. **Base di Conoscenza**
   * **Struttura della Knowledge Base**
     + **Regole**
     + **Fatti**
   * **Categorie Identificate**
   * **Supporto Decisionale**
5. **Dataset e Features**
6. **Modelli di Classificazione**
   * **K-Nearest Neighbors**
   * **GaussianNB**
   * **Random Forest**
   * **Extra Trees**
7. **Analisi dei Grafici di Regressione**
8. **Ottimizzazione degli Iperparametri**
   * **Curva di Validazione**
   * **Exhaustive Grid Search**
9. **Accuratezze dei Classificatori**
   * **Metriche di Valutazione**
   * **Confronto tra Classificatori**
   * **Confusion Matrix Extra Trees Classifier**
10. **Clustering**
    * **Algoritmo KMeans**
    * **Valutazione con Purity**
11. **Sviluppi Futuri**
    * **Integrazione di Algoritmi di Deep Learning**
    * **Miglioramenti nell’Algoritmo di Clustering**
    * **Interfaccia Utente e Funzionalità Interattive**

**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.Linguaggio e Ambienti Utilizzati**

Il progetto è stato sviluppato utilizzando il linguaggio di programmazione Python 3.12.7, scelto per la sua semplicità sintattica, la vasta gamma di librerie disponibili e le solide capacità di elaborazione dei dati e machine learning. Per eseguire il codice e gestire le dipendenze, è consigliato un ambiente virtuale Python per garantire l’isolamento delle librerie necessarie e prevenire conflitti di versione.

Librerie Utilizzate

Nel progetto sono state utilizzate diverse librerie Python per gestire il flusso dei dati, costruire e valutare i modelli di machine learning, e visualizzare i risultati. Ecco le principali librerie impiegate:

* NumPy: utilizzata per la gestione e manipolazione degli array numerici, essenziale per la gestione dei dataset di input.
* Pandas: impiegata per il caricamento e la manipolazione dei dati attraverso dataframe, semplificando operazioni di pre-elaborazione.
* scikit-learn: fondamentale per la costruzione e valutazione dei classificatori (K-Nearest Neighbors, Naive Bayes, Extra Trees, Random Forest), e per strumenti di splitting del dataset come train\_test\_split e classification\_report.
* Matplotlib: utilizzata per la creazione di grafici e la visualizzazione dell’importanza delle feature, consentendo di interpretare visivamente i risultati.
* Custom Libraries (songFeature, classifiers, metrics): librerie personalizzate per la gestione delle caratteristiche delle canzoni, costruzione dei modelli di classificazione e visualizzazione di metriche e matrici di confusione. Questi moduli sono cruciali per l’integrazione dei dati specifici del progetto e la visualizzazione delle performance.

Questo insieme di strumenti consente una gestione efficiente del processo di machine learning e una valida interpretazione dei risultati attraverso grafici e metriche di valutazione.

**3.Manuale d'Uso**

Il programma è progettato per eseguire una serie di classificazioni su caratteristiche di brani musicali utilizzando diversi algoritmi di machine learning. Per utilizzare correttamente il programma, è possibile seguire questi passaggi:

-Preparazione del Dataset: assicurarsi che il file data.csv contenga i dati del brano in formato appropriato, con una colonna label che identifichi la classe e una colonna filename per il nome del file del brano.

-Esecuzione del Programma:

* + Eseguire il file main.py tramite il comando “python main.py” da terminale per caricare il dataset e addestrare diversi classificatori: K-Nearest Neighbors, Naive Bayes, Extra Trees, e Random Forest.
  + Durante l'esecuzione, il programma visualizzerà varie metriche di valutazione e genererà grafici di importanza delle feature per interpretare i risultati.

-Analisi dei Risultati:

* + Il programma genererà una matrice di confusione per ogni classificatore utilizzato, consentendo di analizzare la precisione del modello.

Immagine che contiene testo, schermata, numero, quadrato

Descrizione generata automaticamente Immagine che contiene testo, schermata, quadrato, numero

Descrizione generata automaticamenteImmagine che contiene testo, schermata, numero, quadrato

Descrizione generata automaticamente Immagine che contiene testo, schermata, quadrato, numero

Descrizione generata automaticamente

* + Verranno visualizzati i punteggi di accuratezza per ogni algoritmo e una classificazione dettagliata con il report finale.

-Predizione su Nuovi Brani:

* + L’utente può inserire il percorso di un nuovo brano per ottenere le previsioni tramite i modelli già addestrati.
  + Per farlo, seguire le istruzioni inserendo il percorso del brano al prompt, e il programma restituirà le previsioni di classificazione per ciascun modello.



Immagine che contiene testo, schermata, Carattere

Descrizione generata automaticamente

-Ripetizione delle Predizioni:

* + È possibile effettuare ulteriori predizioni ripetendo il processo per nuovi brani musicali, confermando al prompt.

-Visualizzazione dell’Importanza delle Caratteristiche:

* + Al termine delle predizioni, il programma mostrerà l’importanza delle feature attraverso un grafico, utile per comprendere quali caratteristiche del brano influenzano maggiormente i risultati di classificazione.

**4.Base di Conoscenza**

La base di conoscenza per il sistema di classificazione dei brani musicali è strutturata in modo tale da identificare automaticamente le caratteristiche dei brani e associare ognuno di essi alla classe corretta.

Struttura della Knowledge Base

La base di conoscenza si basa su un sistema di regole, che definiscono come determinati attributi dei brani influenzano l’assegnazione a una specifica categoria. La struttura si compone di regole di classificazione e fatti:

Regole

Le regole di classificazione stabiliscono le condizioni per le varie categorie basate sulle caratteristiche del brano, come il ritmo, la tonalità e altri parametri numerici. Esempi di regole utilizzate sono:

* classical\_genre ⟵ tempo < 60 ∧ harmony = complex
* rock\_genre ⟵ tempo >= 100 ∧ distortion = high
* jazz\_genre ⟵ tempo = medium ∧ improvisation = true
* electronic\_genre ⟵ synth = true ∧ bass = prominent

Ogni regola contribuisce alla classificazione del brano, permettendo al sistema di determinare il genere musicale corrispondente in base alle feature rilevate.

Fatti

I fatti rappresentano le osservazioni e le caratteristiche estratte dai brani musicali e sono suddivisi in due categorie:

* Fatti Dichiarati: Questi sono i dati raccolti dalle caratteristiche dei brani durante la loro analisi, come:
  + tempo: la velocità o il ritmo del brano
  + harmony: livello di complessità armonica
  + distortion: presenza di effetti di distorsione
  + synth: presenza di suoni sintetizzati
  + bass: predominanza delle frequenze basse
  + improvisation: indicazione della presenza di elementi di improvvisazione
* Fatti Derivati: Questi fatti sono il risultato dell’applicazione delle regole di classificazione e rappresentano le categorie di genere determinate dal sistema. Ad esempio:
  + classical\_genre: il brano è classificato come musica classica
  + rock\_genre: il brano è classificato come rock
  + jazz\_genre: il brano è classificato come jazz
  + electronic\_genre: il brano è classificato come elettronica

Categorie Identificate

La knowledge base consente di classificare i brani in varie categorie musicali, come musica classica, rock, jazz ed elettronica. Quando un brano non corrisponde a nessuna categoria definita, il sistema offre la possibilità di classificare manualmente il brano o di aggiungere nuove regole di classificazione.

Supporto Decisionale

In assenza di una classificazione precisa, il sistema suggerisce all'utente di ripetere il test con nuovi parametri, oppure di consultare un esperto per definire un’ulteriore categorizzazione, migliorando così la base di conoscenza e adattandola a nuove esigenze musicali.

**5. 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

**6. 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.

**7.Analisi dei Grafici di Regressione**

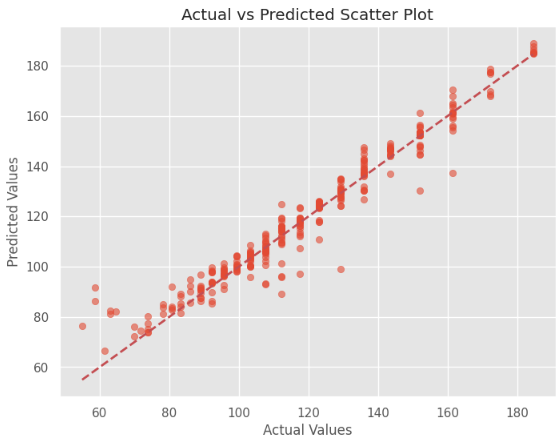
Nei grafici mostrati, viene rappresentato un confronto tra i valori reali (Actual Values) e i valori predetti (Predicted Values) da un modello di regressione applicato su un dataset. Questo tipo di rappresentazione è particolarmente utile per valutare la precisione e l'efficacia dei modelli di regressione nel prevedere risultati continui.

Nel contesto del nostro progetto, questi grafici possono fornire un’indicazione visiva della correlazione tra le caratteristiche dei brani musicali e le rispettive classi o caratteristiche derivate, anche se il nostro sistema è orientato alla classificazione piuttosto che alla regressione. Tuttavia, possiamo applicare un modello di regressione su singole feature numeriche (come `tempo`, `rmse`, o `spectral\_centroid`) per verificare se è possibile predire una caratteristica musicale in base a un'altra, oppure esplorare relazioni complesse tra le feature.

In entrambi i grafici, la linea diagonale rappresenta la "linea ideale" di regressione, che indica dove i punti dovrebbero trovarsi se le previsioni fossero perfette. Maggiore è la vicinanza dei punti a questa linea, migliore è l'accuratezza del modello. L'andamento lineare evidenziato nei grafici suggerisce che il modello di regressione riesce a catturare in modo accettabile la relazione tra i valori, con una dispersione accettabile intorno alla linea di regressione.

Questa rappresentazione potrebbe essere integrata nel nostro progetto per analizzare come diverse caratteristiche audio possano essere predette o correlate tra loro, offrendo un’ulteriore comprensione del comportamento delle feature e supportando la selezione di feature rilevanti per la classificazione.





**8.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.

**9.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.

* Confusion matrix Extra Trees Classifier:

Immagine che contiene quadrato, Rettangolo, pixel

Descrizione generata automaticamente

* Metriche utilizzate Extra Trees Classifier:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Genere | Precision | Recall | F1-score | Support |
| Blues | 0.92 | 0.60 | 0.73 | 20 |
| Classical | 0.91 | 1.00 | 0.95 | 20 |
| Country | 0.46 | 0.30 | 0.36 | 20 |
| Disco | 0.56 | 0.75 | 0.64 | 20 |
| Hiphop | 0.71 | 0.75 | 0.73 | 20 |
| Jazz | 0.61 | 0.70 | 0.65 | 20 |
| Metal | 0.73 | 0.80 | 0.76 | 20 |
| Pop | 0.70 | 0.80 | 0.74 | 20 |
| Reggae | 0.65 | 0.55 | 0.59 | 20 |
| Rock | 0.58 | 0.55 |  |  |

**10.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

**11.Sviluppi Futuri**

Il progetto attuale offre una solida base di classificazione musicale e analisi delle caratteristiche dei brani, ma esistono diverse direzioni attraverso cui potrebbe essere ulteriormente migliorato e espanso per adattarsi a nuove esigenze e tecnologie emergenti.

* Integrazione di Algoritmi di Deep Learning: L’introduzione di reti neurali profonde potrebbe significativamente aumentare l’accuratezza della classificazione dei brani musicali. Modelli come Convolutional Neural Networks (CNN) o Recurrent Neural Networks (RNN) potrebbero essere addestrati per riconoscere pattern complessi nelle tracce audio che i classificatori tradizionali potrebbero non catturare. L’implementazione di tali modelli potrebbe includere l'uso di librerie come TensorFlow o PyTorch e la creazione di un set di dati più ampio e variegato per l’addestramento.
* Miglioramenti nell’Algoritmo di Clustering: Attualmente il sistema utilizza K-Means per il clustering. In futuro, potrebbero essere esplorati algoritmi di clustering più avanzati come DBSCAN o algoritmi basati su modelli misti gaussiani (Gaussian Mixture Models, GMM), che sono in grado di gestire meglio la varietà delle forme dei cluster e la presenza di rumore nei dati. Questo migliorerebbe la capacità del sistema di identificare sotto-gruppi di brani con caratteristiche simili ma non ovvie, migliorando la personalizzazione delle raccomandazioni musicali.
* Interfaccia Utente e Funzionalità Interattive: Lo sviluppo di una interfaccia utente grafica GUI renderebbe il sistema più accessibile agli utenti non tecnici, consentendo loro di caricare brani, visualizzare analisi e ricevere raccomandazioni con facilità. L'aggiunta di funzionalità interattive, come la possibilità per gli utenti di etichettare manualmente i brani e di fornire feedback sulle classificazioni, potrebbe inoltre migliorare l’apprendimento del sistema e la sua precisione attraverso l’apprendimento attivo o il feedback loop.