Music Genre Recognition

Aloia Marcello Giovanni, Belvedere Vincenzo, Leone Marco.

Link al progetto GitHub

1. Introduzione:

Il task di classificazione dei generi musicali, a partire da un determinato file audio (nel caso del progetto, di tipo .WAV), è argomento su cui sono stati svolti numerosi studi in ambito di Machine Learning. Il task consiste nel confronto e nella valutazione di alcuni dei principali modelli di classificazione basati su apprendimento supervisionato. L'Apprendimento Supervisionato è una tecnica di Machine Learning che mira ad istruire un sistema informatico, in modo da consentirgli di elaborare autonomamente previsioni sui valori di output rispetto ad un dato input, sulla base di una serie di esempi ideali costituiti da coppie < Dati, Etichetta >, che vengono inizialmente forniti al modello.

Tale classificazione avverrà sulla base di features estraibili direttamente da un tracce audio, le quali verranno riconosciute tra uno dei 10 generi principali del palinsesto musicale:

- Blues
- Classical
- Country
- Disco
- Hip Hop
- Jazz
- Metal
- Pop
- Reggae
- Rock

Per la rappresentazione dei dati, è stato scelto un modello notoriamente utilizzato nei task di riconoscimento vocale e classificazione di file audio.

I modelli di classificazione confrontati sono stati scelti fra i principali delle seguenti categorie di apprendimento:

- Case Based Reasoning: *K-Nearest Neighbors Classifier*
- Probabilistic Classifiers: *Naïve Bayes Classifier*
- Ensemble Learning Models: *Random Forest Classifier* & *Extra Trees Classifier*

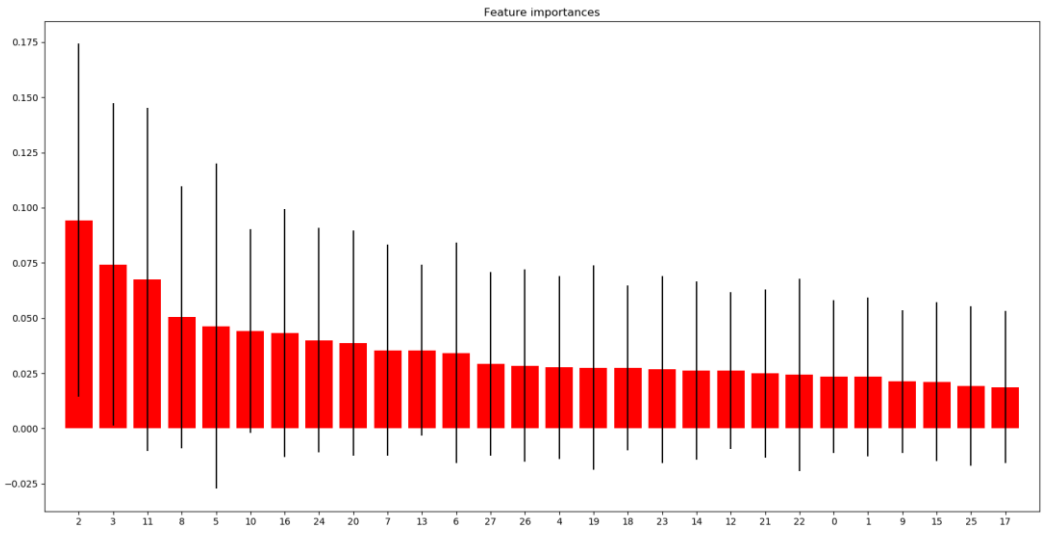
Al termine di tale confronto, è stato adottato il modello più prestante in termini di accuratezza, precisione e richiamo, l'*Extra Trees Classifier*.

2. Dataset e Features:

I dati utilizzati durante il progetto sono stati ottenuti dal dataset *"Music Features"* di Kaggle. Il dataset contiene un totale di 1000 esempi di file audio (suddivisi in 800 per il training e 200 per il test) sottoforma delle specifiche tecniche estrapolate dai file stessi. Le Features derivate da questa estrapolazione sono:

- **Tempo (0):** Misura dell'andamento della musica
- **Beats (1):** Unità ritmica della musica
- **Chroma_stft (2):** Short Time Fourier Transform
- **RMSE (3):** Root Mean Square Error
- **Spectral_centroid (4):** Indica dov'è posizionato il "centro di massa" dello spettro
- **Spectral_bandwith (5):** Misura dell'ampiezza d'onda dello spettro il cui valore non può essere minore della metà del suo massimo
- **Roll-off (6):** L'attenuazione via via crescente delle frequenze gravi o acute man mano che ci si allontana dalla gamma centrale.
- **Zero_crossing_rate (7):** Velocità con cui il segnale cambia da positivo a negativo, o viceversa
- **MFCCs (8-27):** Mel-frequency cepstral coefficients (MFCCs) sono i coefficienti che insieme identificano l'MFC (sono stati presi in considerazione 20 coefficienti)
- **Label (Target Feature):** Nome del genere associato al file audio

Queste features rappresentano le caratteristiche essenziali di un file audio (.WAV nel caso corrente). Attraverso una fase di pre-processing del dataset, si è evinto che, al fine della classificazione, tutte le caratteristiche sono egualmente discriminanti; non è stato quindi possibile ridurre dimensionalmente il numero delle features.



Si è passato dunque a delle tecniche di Feature Extraction in modo da valutare se fosse conveniente ridurre il numero di features a favore dell'accuratezza e/o dell'efficienza di computazione. Le tecniche utilizzate sono state:

- **Principal Component Analysis:**
  PCA è una delle tecniche di riduzione dimensionale lineare più utilizzate. Quando la si utilizza, prendiamo come input i dati originali e si prova a trovare una combinazione delle features che possa riassumere al meglio la distribuzione di questi dati in modo da ridurne le dimensioni. La PCA è in grado di farlo massimizzando le varianze e minimizzando l'errore di ricostruzione osservando le distanze sagge della coppia. Nell'utilizzo della PCA, i dati originali vengono proiettati in una serie di assi ortogonali e ciascuno degli assi viene classificato in ordine di importanza.
- **Linear Discriminant Analysis:**
  La LDA è una generalizzazione del discriminante lineare di Fisher. La LDA funziona creando una o più combinazioni lineari di predittori, creando una nuova variabile latente per ciascuna funzione. Queste funzioni sono chiamate funzioni discriminanti. Il numero di funzioni possibili è uno dei due N_g*-1,* dove N_g= numero di gruppi, o *p*(il numero di predittori), quale dei due sia il più piccolo. La prima funzione creata massimizza le differenze tra i gruppi su quella funzione. La seconda funzione massimizza le differenze su quella funzione, ma non deve essere correlata con la funzione precedente. Questo continua con le funzioni successive con il requisito che la nuova funzione non sia correlata con nessuna delle funzioni precedenti. Quando si utilizza LDA, si presume che i dati di input seguano una distribuzione gaussiana, come nel caso corrente (i grafici di distribuzione dei valori delle feature sono visibili su Kaggle).
- **Independent Component Analysis:**
  L'ICA è un metodo di riduzione della dimensionalità lineare che prende come dati di input una miscela di componenti indipendenti e mira a identificarli correttamente (eliminando tutto il rumore non necessario). Due funzioni di input possono essere considerate indipendenti se la loro dipendenza lineare e non lineare è uguale a zero. L'utilizzo di questa tecnica è motivato da alcune ricerche che hanno mostrato come le componenti dei file audio siano, a gruppi, indipendenti fra loro.

3. Modelli di classificazione utilizzati:

Al fine di ottenere una predizione sui nuovi esempi, sono stati applicati modelli di classificazione basati su apprendimento supervisionato, derivati dalla libreria *sklearn*. L'idea di utilizzare più modelli ha avuto lo scopo di valutare l'accuratezza di ogni singolo modello in fase di test.

- **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:

$$P(y \mid x_1, \ldots, x_n) = \frac{P(y)P(x_1, \ldots, x_n \mid y)}{P(x_1, \ldots, x_n)}$$

  Dove $P(y|x_1,\ldots,x_n)$ è la probabilità a posteriori, $P(y)$ è la probabilità a priori, $P(x_1,\ldots,x_n|y)$ è la verosimiglianza e $P(x_1,\ldots,x_n)$ è la funzione di partizione. Nell' utilizzo del classificatore GaussianNB, si presume che la probabilità delle feature sia gaussiana:

$$P(x_i \mid y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} \exp\left(-\frac{(x_i - \mu_y)^2}{2\sigma_y^2}\right)$$

  I parametri σ_y e μ_y 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.

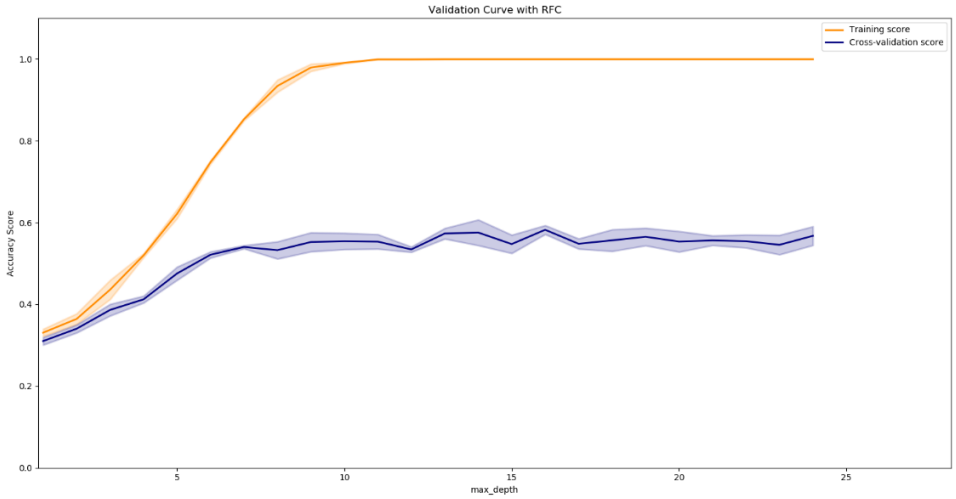
- **Extra Trees:**
  Tale modello è simile al precedente, la differenza risiede nella scelta degli alberi, la quale avviene in maniera puramente casuale.

3.1. Ottimizzazione degli iperparametri:

Al fine di rendere notevolmente alta l'accuratezza di ciascun classificatore utilizzato è stato seguito un procedimento di ottimizzazione degli iperparametri. Partiamo dal presupposto che ciascun modello di classificazione prevede la presenza di parametri opportunamente passati in fase di costruzione di un determinato modello, dunque ciascun valore associato a questi ultimi prendi il nome di iperparametro. Se non esplicitati, ai parametri verranno associati valori di default, che molto spesso non permettono al modello di esaltare la sua massima accuratezza.

Dunque, qualsiasi parametro di un qualsiasi classificatore, può essere opportunamente settato sulla base di diversi approcci di ottimizzazione. Le tecniche di ottimizzazione utilizzate in tale progetto sono state:

- **Curva di validazione:**
  Tale metodo è utile al fine di verificare visivamente i valori potenzialmente ottimizzati di ciascun modello. Una curva di validazione può essere tracciata su un grafico, per mostrare come un modello si comporta con diversi valori di un singolo iperparametro.



Attraverso questo grafico, possiamo notare come avviene tale procedimento e sulla base di quale metrica si stabilisce il giusto settaggio di un iperparametro. Tale grafico riporta la curva di validazione per il parametro *"max_depth"* presente nell'ExtraTrees Classifier;

sull'asse delle ordinate è presente il valore di accuratezza, metrica fondamentale ai fini dell'utilizzo di tale procedura, mentre sull'asse delle ascisse abbiamo il parametro che intendiamo settare sulla base di diversi valori che può avere.

Infine, le due curve (training score, cross-validation score) rappresentano il vero e proprio concetto principale di tale procedura. In base alle stesse di può controllare quale è il valore, dell'iperparametro, per cui l'accuratezza diventa massima.

- **Exhaustive grid search:**

  Tale metodo, fornito da *GridSearchCV,* genera in maniera esaustiva i possibili candidati (iperparametri) attraverso una griglia di valori specificata opportunamente dal parametro *"param_grid",* caratterizzato da un range di valori per ogni singolo parametro specificato dall'utente. In maniera del tutto automatica, vengono valutate tutte le possibili combinazioni di assegnazioni degli iperparametri e viene mantenuta la combinazione migliore. Al termine di tale processo, verranno mostrati quelli che sono gli iperparametri migliori per un determinato modello di classificazione.

  Delle tante procedure utili ai fini di tale topic, sono state scelte proprio queste due in quanto la prima si basa su un procedimento del tutto "manuale" e fortemente esplicativo, visto l'utilizzo di un grafico, il secondo invece è del tutto "automatico",

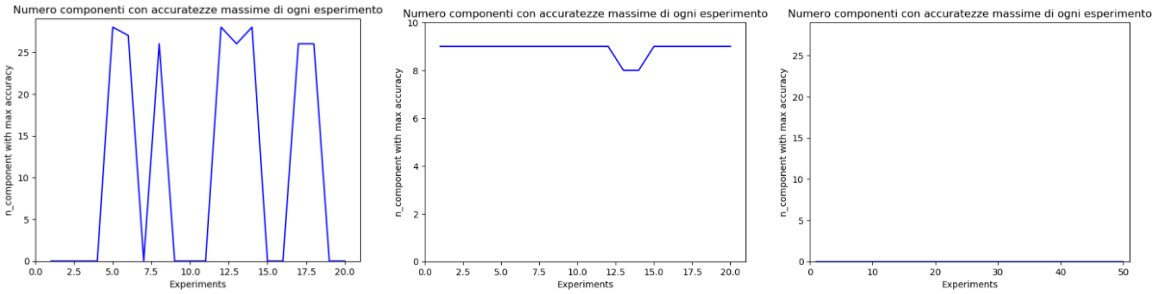
tentando ogni combinazione, sulla base dell'accuratezza raggiunta in ogni singolo tentativo.

Ai seguenti link, il riferimento a tutti i grafici ottenuti: ExtraTrees Random Forest

4. Risultati e Discussione:

4.1. Sperimentazioni Features Extraction

Passando alle tecniche di Features Selection si è evinto che le features all'interno del dataset erano tutte egualmente rilevanti al fine della classificazione e che c'era già stata una fase di pre-processing, la quale ci ha permesso di non dover lavorare su un dataset che comprendesse direttamente file musicali da pre-processare e da cui identificare le feature più discriminanti nell'analisi di file audio. Ciò si può denotare dai risultati ottenuti dalle sperimentazioni effettuate su ogni tecnica di Feature Extraction: per ogni tecnica sono state svolte 20 sperimentazioni, da ognuna delle quali è stata recuperata l'accuratezza massima; tale accuratezza è stata calcolata per tutti i possibili valori del parametro *n_features*.



Convenzionalmente, assumiamo che *n_features = 0* sia il caso di sperimentazione in cui la tecnica di Feature Extraction non sia stata utilizzata. Tutte le sperimentazioni sono state effettuate su un ExtraTrees Classifier con i parametri impostati sui valori precedentemente valutati come ottimali.

Nel primo grafico si nota l'andamento altalenante delle accuratezze massime ottenute nelle sperimentazioni dell'utilizzo della PCA: si nota come, in maniera totalmente dipendente dallo split del dataset, i valori massimi di accuratezza siano ottenuti o nel caso in cui la PCA non venisse utilizzata o nel caso in cui il numero di feature fosse almeno 27.

Nel secondo grafico è riportato l'andamento delle sperimentazioni sull'utilizzo della LDA, il quale dimostra come le accuratezze massime siano ottenute nel momento in cui *n_features* = 9, nella maggior parte dei casi. Si è constatato tuttavia, che tali accuratezze non siano veritiere, dal momento che, utilizzando un validation set, le predizioni abbiano restituito dei risultati insoddisfacenti (le predizioni risultavano errate e con probabilità massima non superiore a 0.2).

Nel terzo grafico si evince come in tutte le sperimentazioni l'accuratezza massima sia ottenuta quando la tecnica di ICA non viene utilizzata; tale tecnica è stata dunque scartata a priori.

Ai seguenti link sono riportati i risultati di ogni singola sperimentazione su tutte le tecniche: PCA ICA LDA

4.2 Accuratezze classificatori

Dopo aver eseguito il tuning dei diversi iperparametri di ciascun classificatore, essi sono stati utilizzati al fine di predire la giusta classe di una determinata preview (anteprima file audio). Di seguito vengono riportati tutti i valori delle metriche utilizzate al fine di valutare la "bontà" di ciascuno dei classificatori. Tutti i classificatori, sulla base di ciò che viene detto nella sezione precedente, sono stati addestrati su dati non sottoposti a tecniche di feature extraction, perché inefficienti al fine del task.

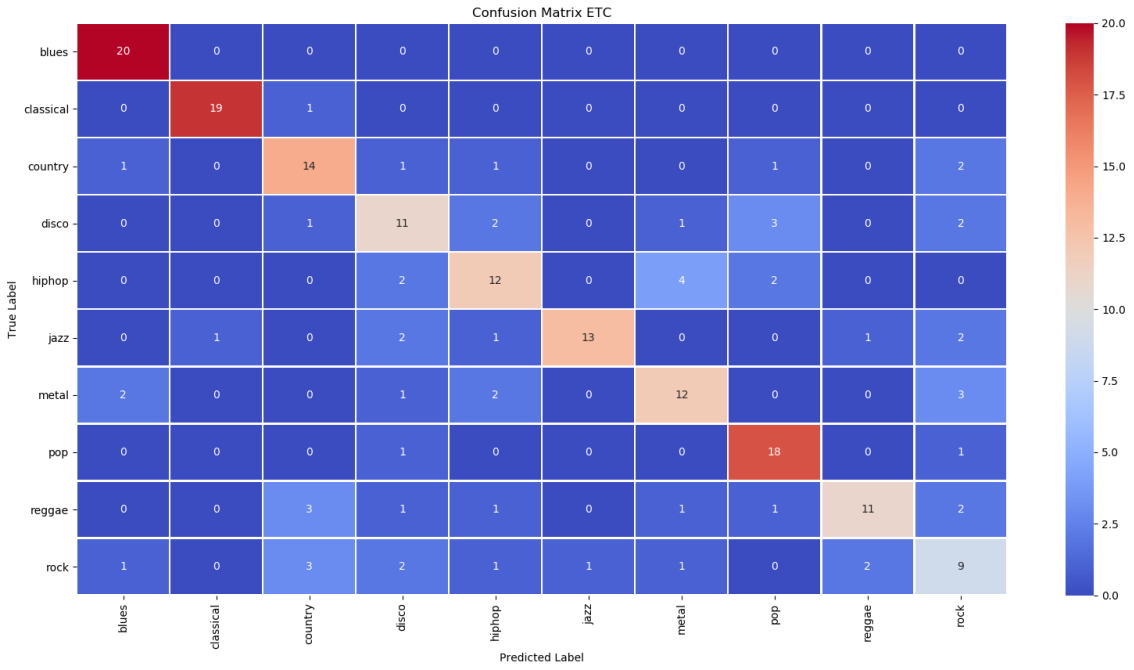
Classificatori	Accuracy	Precision	Recall	F1-measure
K-NN	0.365	0.371	0.365	0.356
GaussianNB	0.435	0.404	0.435	0.389
RandomForest	0.530	0.535	0.530	0.505
ExtraTrees	0.670	0.665	0.670	0.665

5. Conclusione:

Dopo aver confrontato i diversi classificatori, si è arrivati alla conclusione di utilizzare l'ExtraTreesClassifier. Tale scelta si è basata sulle metriche sopra riportate. Esso, visto il valore più alto di accuratezza, riesce a predire meglio quale sarà il genere di una determinata preview musicale.

Di seguito vengono riportate:

- **Confusion matrix Extra Trees Classifier:**



- **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	0.56	20

6. Contatti:

- Aloia Giovanni Marcello 665275 : giovanni.marcello.aloia@gmail.com
- Belvedere Vincenzo 675996 : vincenzo.belvedere.99@gmail.com
- Leone Marco 690135 : marco_leone_96@hotmail.it