CORSO DI INGEGNERIA DELLA CONOSCENZA

RELAZIONE PROGETTO

FEEL THE MUSIC:

WHAT DO YOU "NEED" TO LISTEN?

Studentessa: Nido Marianna, *Matricola*: 681041, *E-mail*: m.nido@studenti.uniba.it

Link al repository GitHub: https://github.com/MariannaNido/FeelTheMusic

INDICE

1. Introduzione
2. Il Dataset
   - 2.1 Descrizione
   - 2.2 Modifiche al Dataset
3. Il Questionario
4. Strumenti Utilizzati
   - 4.1 k-Nearest Neighbors
   - 4.2 Gaussian Naïve Bayes
   - 4.3 Decision Tree
   - 4.4 Support Vector Machine
   - 4.5 Random Forest
5. Architettura del sistema
6. Valutazioni
   - 6.1 k-Nearest Neighbors (vicini = 5)
   - 6.2 Gaussian Naïve Bayes
   - 6.3 Decision Tree (limite profondità = 5)
   - 6.4 Support Vector Machine
   - 6.5 Random Forest (limite profondità = 10)

1. Introduzione

Feel The Music è un progetto sviluppato per il Corso di Ingegneria della Conoscenza dell'Università degli Studi di Bari Aldo Moro, Dipartimento di Informatica.
Lo scopo principale di questo progetto è suggerire all'utente un genere musicale che si avvicini di più al suo "stato" attuale, e che quindi si spera faccia in modo che la musica che sceglierà di ascoltare sia accurata e di giusta compagnia in base a qualunque cosa stia provando o stia per fare.

All'utente verrà somministrato un questionario, che aiuterà alla scelta del genere.

Un punto che si vuole legare allo sviluppo di questo semplice progetto sta anche nel fare in modo che l'utente possa aprirsi a generi musicali diversi da quelli a cui è abituato, e quindi scoprire nuovi stili e ampliare l'eventuale bagaglio musicale già posseduto.

2. Il Dataset

Il dataset originale utilizzato per questo progetto è stato ricavato da *Kaggle*, e contiene un elenco di 2664 generi musicali estratti da Spotify, ognuno di essi descritti da 14 features. Il dataset lo si può trovare già all'interno del package datasetUtils del progetto.

Link per il download del dataset: https://www.kaggle.com/datasets/pesssinaluca/spotify-by-generes

2.1 Descrizione

Di seguito, un estratto del dataset principale che mostra anche le caratteristiche rappresentate:

genres	acousticness	danceability	duration_ms	energy	instrumentalness	liveness
2664 unique values	0 – 1	0.06 – 0.96	30.9k – 1.64m	0 – 1	0 – 0.99	0.02 – 0.93
acoustic pop	0.48524972623281115	0.5278208104445249	234521.88943792926	0.4829903476508946	0.04711369348162551	0.15336058534714428
afghan pop	0.45039999999999997	0.6968333333333333	301078.0833333333	0.6721666666666666	0.0014728504166666666	0.09867083333333333
afropop	0.4415449363689098	0.6084024753946681	323634.7059977444	0.5253437564719334	0.13170376170997292	0.15768562389437807
albanian pop	0.09593499999999999	0.6935	167717.5	0.782	1.215e-05	0.15595
alternative pop	0.10237811607892443	0.4772272546962806	217119.0054985515	0.7436669747758774	0.1543763411821494	0.17597202176447413
alternative pop rock	0.18306958333333326	0.5961510416666668	208448.07118055553	0.7148621527777778	1.1372309027777777e-05	0.19344538194444447
ambient pop	0.7401253333333333	0.32154000000000005	228714.02666666667	0.25361999999999996	0.7206852000000001	0.10936666666666665
antiviral pop	0.17394746827190827	0.6351263016263015	201968.43857493857	0.7202116707616709	0.016358727689013686	0.21266052942552943

loudness	speechiness	tempo	valence	popularity	key	mode
-37.6 – -0.13	0.02 – 0.94	37.8 – 210	0 – 0.98	0 – 82	0 – 11	0 – 1
-9.214582532432132	0.040121910021044496	117.5559650721883	0.38460085043698067	52.86698245963172	11	1
-7.560708333333334	0.08020000000000001	113.18816666666667	0.8082916666666667	38.583333333333336	7	1
-11.340903428446786	0.07655866305436904	115.07851337323159	0.6300199358919044	43.583738488348125	7	1
-7.0075	0.11395000000000001	125.498	0.6215	56.5	6	0
-8.96483470672043	0.04899254407095954	128.1240144133023	0.5720638086530863	37.856017486308644	11	1
-4.8288947916666665	0.06517465277777779	119.6851767361111	0.6104333333333333	53.50000000000001	10	1
-16.705946666666666	0.03571066666666667	95.59444666666667	0.181174	42.56	7	1
-6.953785845910846	0.11023041154791156	130.28318356148358	0.6261972885222886	53.2747425997426	2	1

Le caratteristiche per ogni genere musicale sono 14, e sono le seguenti:

Feature	Descrizione
Genres	genere musicale
Acousticness	probabilità che la traccia sia acustica o meno [0,1]
Danceability	probabilità che la traccia sia adatta o meno al ballo [0,1]
Duration_ms	durata della traccia, in millisecondi (ms)
Energy	probabilità che la traccia sia intensa e attiva [0,1]
Instrumentalness	probabilità che la traccia contenga o meno voce [0,1]
Liveliness	presenza di pubblico nella traccia [0,1]
Loudness	volume complessivo della traccia in decibel (dB). I valori tipici sono compresi tra -60 e 0 dB, ma il dataset inizia ad avere dei valori da -40.
Speechiness	probabilità che la traccia contenga parole pronunciate, non cantate [0,1]
Tempo	tempo complessivo stimato di una traccia in battiti al minuto (BPM)
Valence	positività musicale veicolata dalla traccia [0,1]
Popularity	popolarità del brano [0,100]
Key	chiave musicale in cui si trova la traccia. Gli interi vengono mappati alle altezze utilizzando la notazione Pitch Class standard
Mode	modalità (maggiore o minore) di un brano, il tipo di scala da cui deriva il suo contenuto melodico. Maggiore è rappresentato da 1 e Minore 0

2.2 Modifiche al Dataset

Delle 14 features del dataset originale, si è scelto di concentrarsi solo su 9: Genres, Acousticness, Danceability, Energy, Instrumentalness, Loudness, Speechiness, Valence, Popularity.

Dato che alcuni generi sembrano far capo ad un unico macro-genere, e dato che alcuni generi non sono molto conosciuti, si è deciso di modificare il dataset di partenza, categorizzando i generi e raggruppandoli in modo molto semplice in base ad una parola chiave identificativa del genere più generale.

Nello specifico, sono stati estratti e categorizzati i seguenti generi: Classical, Comic Sketch, Country, Dance, Hip Hop, Indie, Jazz, Lo-Fi, Metal, Pop, Rap/Trap, Reggae/Reggaeton, Rock, Techno.

Inoltre, dato che il genere verrà fornito all'utente solo dopo che avrà compilato un questionario, i valori delle 9 features scelte sono stati a loro volta modificati, in modo da ottenere una sorta di standardizzazione dei valori.

Per fare questo, sono state individuate 5 categorie in cui suddividere i valori: Very Low, Low, Medium, High, Very High.

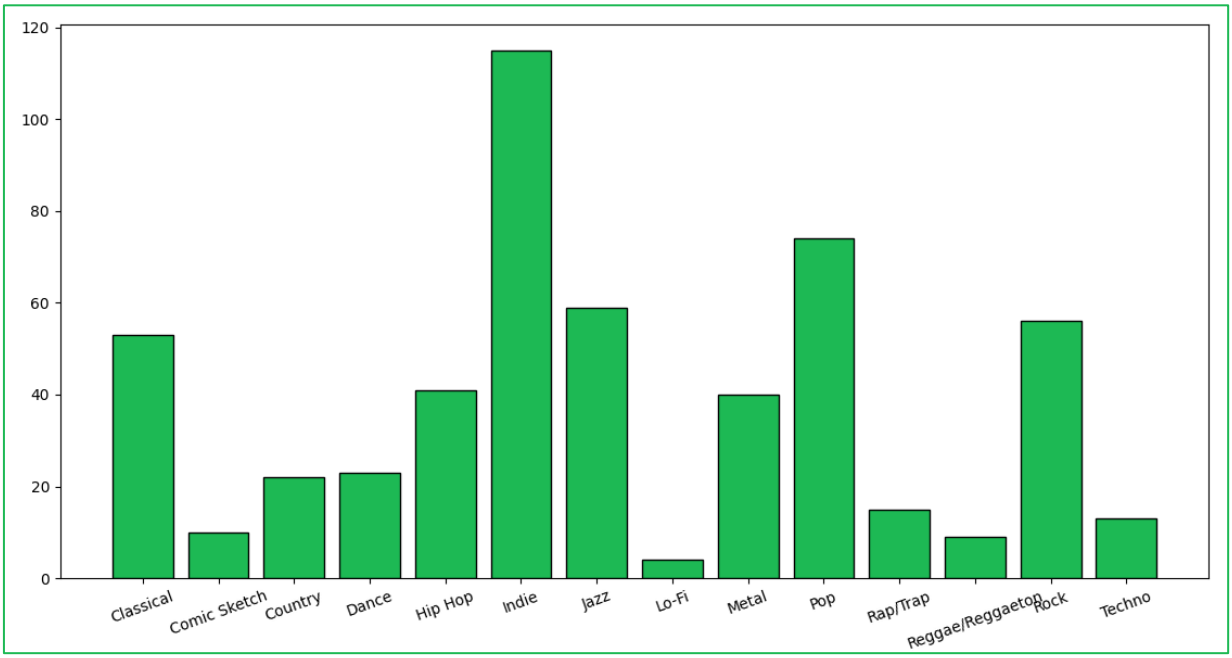
Di seguito, nello specifico, gli intervalli di valori che rientrano nelle categorie appena elencate:

```
Very Low  -> 1 -> [0.0, 0.2) / [ 0, 20) per 'popularity' / [-40, -32) per 'loudness'
Low       -> 2 -> [0.2, 0.4) / [20, 40) per 'popularity' / [-32, -24) per 'loudness'
Medium    -> 3 -> [0.4, 0.6) / [40, 60) per 'popularity' / [-24, -16) per 'loudness'
High      -> 4 -> [0.6, 0.8) / [60, 80) per 'popularity' / [-16, -8) per 'loudness'
Very High -> 5 -> [0.8, 1.0] / [80, 100] per 'popularity' / [-8, 0] per 'loudness'
```

Per evitare la presenza di valori ripetuti, che avrebbero potuto dare problemi in caso di classificazione, è stata adottata la rimozione di duplicati prima per quanto riguarda la riduzione effettuata per i generi, poi per quella riguardante la standardizzazione dei valori.

Queste operazioni di modifica al dataset hanno anche fatto in modo di ridurre il volume dei dati, che passa da un totale di 2664 valori ad un totale ridotto di 534.

Dopo le modifiche, il dataset risulta così composto:



Possiamo notare come il dataset sia leggermente sbilanciato per quanto riguarda alcune classi identificate, in particolare per la classe "Lo-Fi", che ha solo 4.

3. Il Questionario

Il questionario da somministrare all'utente è composto da 8 domande, ognuna avente 5 possibili risposte, numerate da 1 a 5. Ogni domanda si riferisce ad una feature diversa, esclusa la feature "Genres", e in base alle risposte fornite dall'utente verrà creata una lista di valori, che permetterà al programma di effettuare una predizione in base agli elementi contenuti nel dataset modificato.

La lista di risposte dell'utente verrà trattato come se fosse un elemento appartenente al dataset, ma non ancora classificato, e dunque l'obiettivo sarà quello di ottenere il macro-genere che più si avvicina ai valori forniti. Un esempio:

```
INIZIA IL QUESTIONARIO

A. Di solito, che brani preferisci?
    1. Brani molto caotici
    2. Brani movimentati
    3. Una via di mezzo tra il movimentato e il tranquillo
    4. Brani tranquilli
    5. Brani completamente acustici

>
```

4. Strumenti Utilizzati

Il linguaggio utilizzato per sviluppare il progetto è il Python, avendo molte librerie utili per manipolare i dati e applicare algoritmi di machine learning. L'IDE utilizzato è PyCharm.

Librerie principali utilizzate sono Pandas per la gestione del dataset, Matplotlib e Seaborn per la creazione e gestione dei grafici utili e Sklearn per la parte relativa alla classificazione e valutazione dei modelli.

Per l'addestramento dei classificatori, il dataset è stato diviso in features di input e feature di output, individuando la feature "Genres" come feature di output e le restanti come features di input.

I classificatori utilizzati per effettuare la corretta predizione sono: k-Nearest Neighbors, Gaussian Naive Bayes, Decision Tree, Support Vector Machine e Random Forest.

4.1 k-Nearest Neighbors

Il k-Nearest Neighbors è un algoritmo di case-based reasoning, qui utilizzato per un task di classificazione. In questo tipo di modelli, gli esempi di training vengono immagazzinati per poi essere ritrovati nella soluzione di problemi.

Nello specifico, nel k-NN, dato il nuovo esempio, verranno usati i k esempi di training più simili ad esso per predirne il valore target. La predizione può essere la moda, la media o l'interpolazione tra i k esempi di training.

Dunque, nella classificazione con k-NN, l'output è la classe di appartenenza. La classificazione di un oggetto avviene tramite l'assegnazione dello stesso alla classe più comune tra i suoi k vicini.

4.2 Gaussian Naïve Bayes

Il Gaussian Naïve Bayes è un algoritmo di apprendimento supervisionato che fa parte della famiglia dei classificatori probabilistici basati sull'applicazione del teorema di Bayes.

Il caso base di questi classificatori è il Naïve Bayes, ed essi sono caratterizzati dal fatto di essere in grado di fare predizioni su una classe in senso probabilistico, cioè calcolando la probabilità di appartenenza ad una classe, facendo assunzioni di indipendenza condizionata reciproca delle feature di input data la classificazione. È ottimale quando c'è una sola feature osservata, ma al crescere del numero di feature osservate, l'accuratezza dipende dalla reciproca indipendenza delle feature di input, data quella di output.

Nello specifico, il Gaussian Naïve Bayes sfrutta la funzione di densità di probabilità gaussiana per ricavare le varie probabilità di classe, e il valore di

probabilità di classe più alto ottenuto rappresenterà la classe da associare al nuovo esempio da categorizzare.

4.3 Decision Tree

Il Decision Tree è un algoritmo di apprendimento automatico supervisionato, qui utilizzato per un task di classificazione (Classification Tree).
Un Decision Tree è un albero i cui nodi non foglia sono etichettati come le feature di input, e gli archi uscenti da questi nodi sono etichettati con ognuno dei possibili valori delle feature. I nodi foglia sono etichettati con la classe.

Per classificare un nuovo esempio, questo viene "filtrato" attraverso tutto l'albero: per ogni feature, viene seguito l'arco corrispondente al valore; nel momento in cui viene raggiunta una foglia, viene restituita la classe corrispondente all'etichetta della foglia raggiunta.

Nel nostro caso, verrà preso in considerazione il Classification Tree con limite massimo di profondità 5.

4.4 Support Vector Machine

Il Support Vector Machine è un modello lineare di apprendimento automatico supervisionato, qui utilizzato per un task di classificazione (Support Vector Classifier, il cui obiettivo è quello di trovare un iperpiano in uno spazio n-dimensionale, dove n è il numero di features, che classifichi distintamente i dati. L'obiettivo è quello di trovare l'iperpiano che massimizza il margine, e cioè la distanza minima degli esempi dall'iperpiano.

Questa forma di apprendimento si dice non parametrica, nel senso che prescinde dell'uso dei parametri dello spazio, cioè del numero di dimensioni dello spazio, perché dà maggior peso al numero degli esempi più importanti, cioè il numero dei vettori di supporto che vanno a determinare il miglior modello.

4.5 Random Forest

Il Random Forest è un algoritmo di bagging ensemble learning, la cui idea di base è quella di combinare più alberi di decisione, che vengono addestrati separatamente su parti diverse del dataset.
Ognuno degli alberi farà poi la sua predizione su un nuovo esempio, e la predizione finale può essere ottenuta come media delle singole predizioni.

Nello specifico, utilizzeremo un Random Forest Classifier con limite di profondità pari a 10.

5. Architettura del sistema

Il sistema è diviso in 5 package:

- *datasetUtils*, in cui troviamo il file *datasetHandler.py* che contiene tutte le funzioni necessarie alla gestione del dataset originale e creazione del dataset effettivo

utilizzato; inoltre, all'interno del package troviamo il dataset originale *data_by_genres.csv*, e successivamente (dopo la prima esecuzione del programma) anche il dataset modificato, *excerpt_data_by_genres.csv*)

- *surveyUtils*, in cui troviamo il file *surveyHandler.py*, che contiene il questionario da somministrare all'utente
- *classificationUtils*, in cui troviamo il file *classifiersHandler.py*, che contiene le funzioni utili alla classificazione e alla valutazione dei modelli
- *figureUtils*, in cui troviamo il file *figureHandler.py*, che contiene le funzioni utili alla creazione di grafici
- *mainPackage*, in cui troviamo il file *feelMain.py*, che rappresenta il main del progetto, e quindi si occupa di utilizzare le librerie necessarie per far funzionare tutto il programma; nello specifico, richiamerà le funzioni contenute nei file datasetHandler.py, surveyHandler.py, classifiersHandler.py

6. Valutazioni

Per valutare le prestazioni dei vari classificatori utilizzati, si è fatto uso della funzione *StratifiedKFold()*, una variante della k-fold cross validation molto utile in caso di dataset sbilanciati. Questa variante di cross validation ritorna dei set contenenti approssimativamente la stessa percentuale di esempi per ogni classe target del set completo. In questo caso, k = 3.
Inoltre, le predizioni per valutare i modelli sono state effettuate tramite la funzione *cross_val_predict()*, che divide i dati in base al parametro di cross validation scelto. Ogni campione appartiene esattamente ad un insieme di test, e la sua predizione viene calcolata col modello passato.

Nello specifico, verranno visualizzati il *classification report*, la *confusion matrix* (riportate sotto), e singolarmente per ogni modello:

- *Accuracy Score*: numero di corrette predizioni fratto il totale delle predizioni effettuate. Se moltiplicato per 100, diventa una misura percentuale.

$$\frac{TP + TN}{(TP + TN + FP + FN)}$$

  Essendo il nostro dataset leggermente sbilanciato, è opportuno avvalersi anche di ulteriori metriche per valutare il modello.

- *Precision Score*: numero di predizioni positive fratto il numero totale di predizioni positive [Positive Predictive Value].

$$\frac{TP}{TP + FP}$$

  Ci dice quanto il classificatore è stato accurato.

- *Recall Score*: numero di predizioni positive fratto il numero totale di campioni positivi nel dataset [Sensitivity or True Positive Rate].

$$\frac{TP}{TP + FN}$$

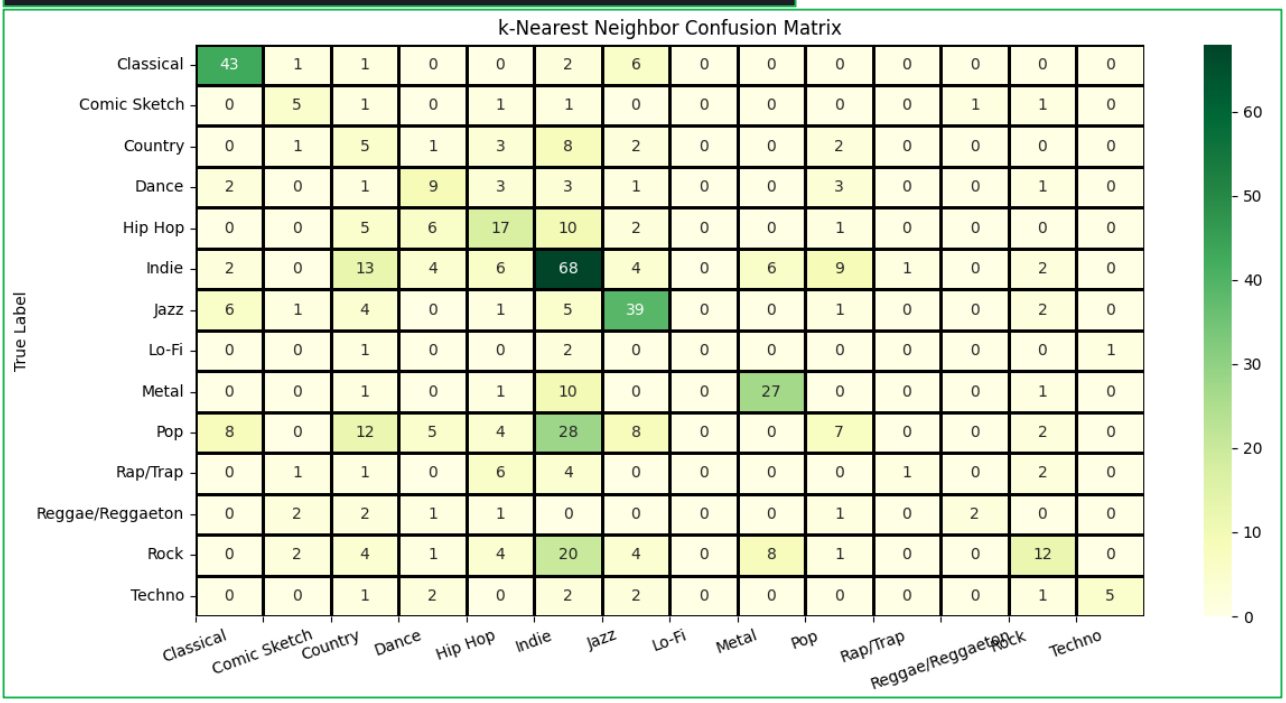
Ci dice quanto il classificatore è stato completo.

- *F1 Score*: valore che mette insieme precision e recall.

$$\frac{2 \cdot (P \cdot R)}{(P + R)}$$

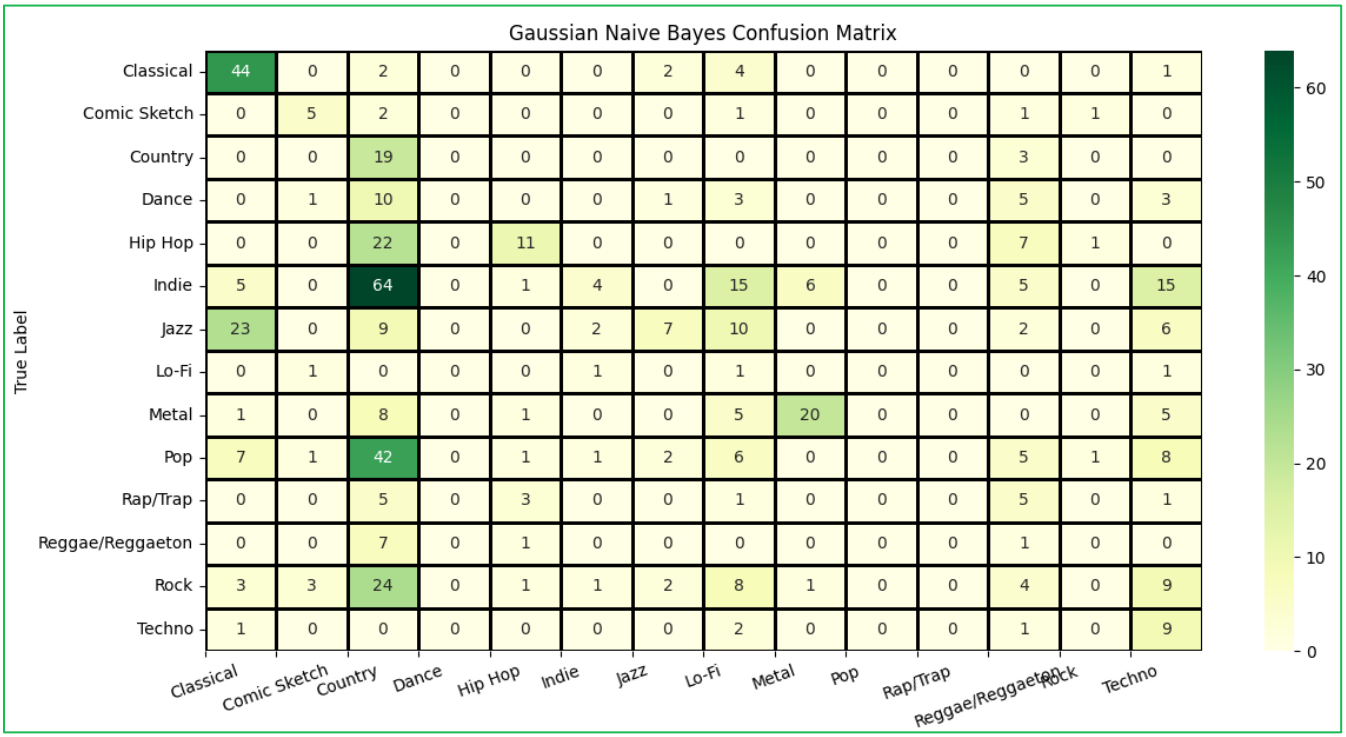
6.1 k-Nearest Neighbors (vicini = 5)

```
### k-Nearest Neighbor Classification report ###

                precision    recall  f1-score   support

      Classical       0.70      0.81      0.75        53
   Comic Sketch       0.38      0.50      0.43        10
        Country       0.10      0.23      0.14        22
          Dance       0.31      0.39      0.35        23
        Hip Hop       0.36      0.41      0.39        41
          Indie       0.42      0.59      0.49       115
           Jazz       0.57      0.66      0.61        59
          Lo-Fi       0.00      0.00      0.00         4
          Metal       0.66      0.68      0.67        40
            Pop       0.28      0.09      0.14        74
       Rap/Trap       0.50      0.07      0.12        15
Reggae/Reggaeton       0.67      0.22      0.33         9
           Rock       0.50      0.21      0.30        56
         Techno       0.83      0.38      0.53        13

       accuracy                           0.45       534
      macro avg       0.45      0.38      0.37       534
   weighted avg       0.46      0.45      0.43       534
```



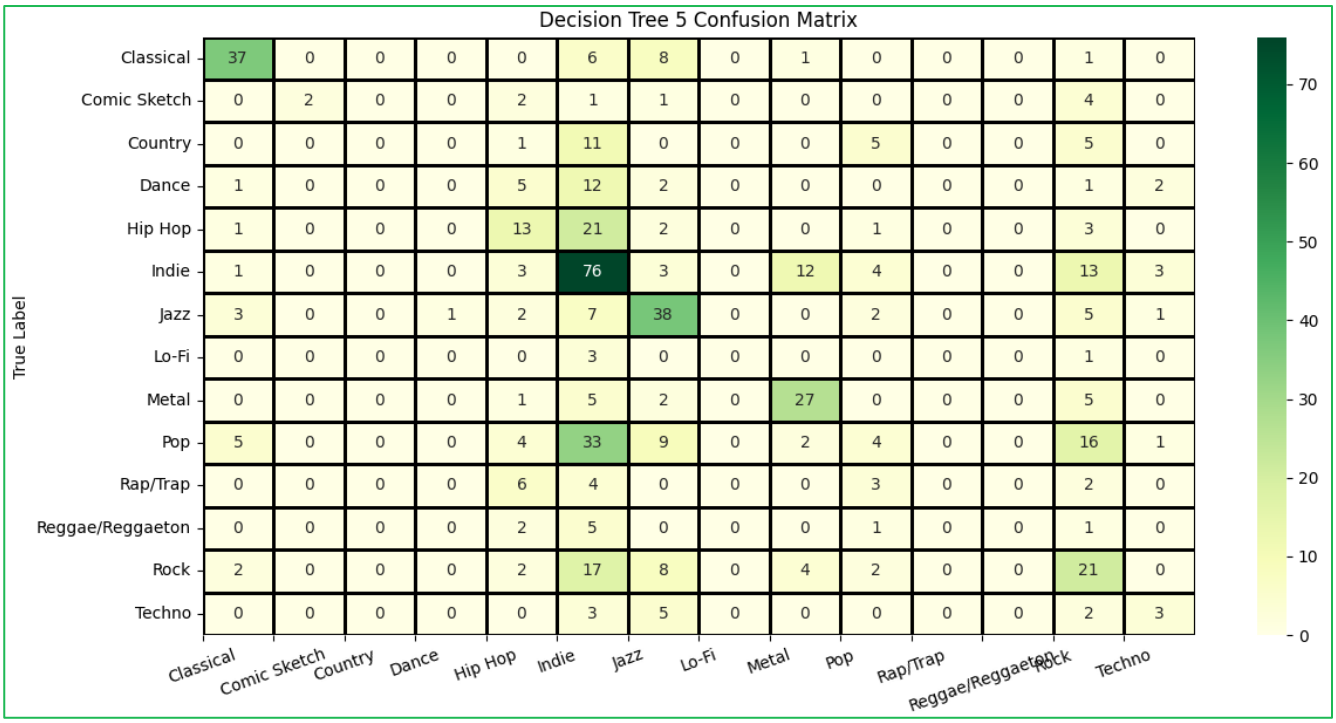
6.2 Gaussian Naïve Bayes

```
### Gaussian Naive Bayes Classification report ###

                precision    recall  f1-score   support

      Classical       0.52      0.83      0.64        53
   Comic Sketch       0.45      0.50      0.48        10
        Country       0.09      0.86      0.16        22
          Dance       0.00      0.00      0.00        23
        Hip Hop       0.58      0.27      0.37        41
          Indie       0.44      0.03      0.06       115
           Jazz       0.50      0.12      0.19        59
          Lo-Fi       0.02      0.25      0.03         4
          Metal       0.74      0.50      0.60        40
            Pop       0.00      0.00      0.00        74
        Rap/Trap       0.00      0.00      0.00        15
Reggae/Reggaeton       0.03      0.11      0.04         9
           Rock       0.00      0.00      0.00        56
         Techno       0.16      0.69      0.25        13

       accuracy                           0.23       534
      macro avg       0.25      0.30      0.20       534
   weighted avg       0.32      0.23      0.19       534
```



6.3 Decision Tree, (limite profondità = 5)

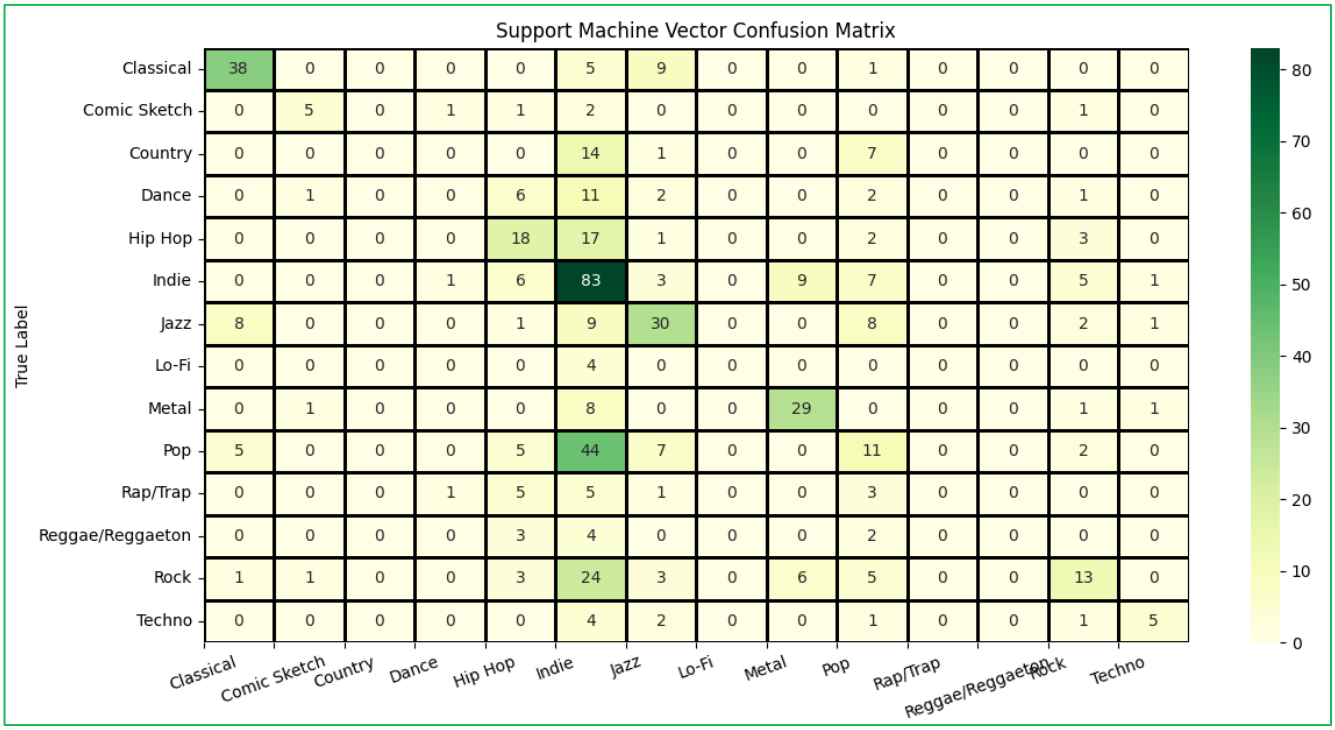
```
### Decision Tree 5 Classification report ###

                precision    recall  f1-score   support

      Classical       0.74      0.70      0.72        53
   Comic Sketch       1.00      0.20      0.33        10
        Country       0.00      0.00      0.00        22
          Dance       0.00      0.00      0.00        23
        Hip Hop       0.32      0.32      0.32        41
          Indie       0.37      0.66      0.48       115
           Jazz       0.49      0.64      0.55        59
          Lo-Fi       0.00      0.00      0.00         4
          Metal       0.59      0.68      0.63        40
            Pop       0.18      0.05      0.08        74
       Rap/Trap       0.00      0.00      0.00        15
Reggae/Reggaeton       0.00      0.00      0.00         9
           Rock       0.26      0.38      0.31        56
         Techno       0.30      0.23      0.26        13

       accuracy                           0.41       534
      macro avg       0.30      0.28      0.26       534
   weighted avg       0.35      0.41      0.36       534
```



6.4 Support Vector Machine

Support Machine Vector Classification report

	precision	recall	f1-score	support
Classical	0.73	0.72	0.72	53
Comic Sketch	0.62	0.50	0.56	10
Country	0.00	0.00	0.00	22
Dance	0.00	0.00	0.00	23
Hip Hop	0.38	0.44	0.40	41
Indie	0.35	0.72	0.48	115
Jazz	0.51	0.51	0.51	59
Lo-Fi	0.00	0.00	0.00	4
Metal	0.66	0.72	0.69	40
Pop	0.22	0.15	0.18	74
Rap/Trap	0.00	0.00	0.00	15
Reggae/Reggaeton	0.00	0.00	0.00	9
Rock	0.45	0.23	0.31	56
Techno	0.62	0.38	0.48	13
accuracy			0.43	534
macro avg	0.33	0.31	0.31	534
weighted avg	0.39	0.43	0.39	534



6.5 Random Forest (limite profondità = 10)

```
### Random Forest 10 Classification report ###

                  precision    recall  f1-score   support

       Classical       0.79      0.77      0.78        53
    Comic Sketch       0.80      0.40      0.53        10
         Country       0.29      0.18      0.22        22
           Dance       0.44      0.30      0.36        23
         Hip Hop       0.39      0.44      0.41        41
           Indie       0.40      0.56      0.47       115
            Jazz       0.59      0.69      0.64        59
           Lo-Fi       0.00      0.00      0.00         4
           Metal       0.64      0.70      0.67        40
             Pop       0.24      0.15      0.18        74
         Rap/Trap       0.33      0.13      0.19        15
Reggae/Reggaeton       0.33      0.11      0.17         9
            Rock       0.31      0.34      0.32        56
          Techno       0.50      0.38      0.43        13

        accuracy                           0.46       534
       macro avg       0.43      0.37      0.38       534
    weighted avg       0.45      0.46      0.44       534
```