Relazione Caso di Studio
Ingegneria Della Conoscenza
Playlist Recommender

Matteo Esposito mat. 718240 m.esposito143@studenti.uniba.it
Giuseppe Galgano mat. 717510 g.galgano7@studenti.uniba.it

Repository GitHub:
https://github.com/espositic/playlist_recommender-ICON

03 Novembre 2022

Contenuti

1 Introduzione

Per il caso di studio abbiamo optato per la realizzazione di un recommender system di canzoni basato su un dataset di Spotify acquisito dal sito Kaggle. Il dataset in questione contiene circa 230000 canzoni ed un totale di 26 generi.

1.1 Strumenti

Abbiamo deciso di utilizzare come linguaggio Python (www.Python.org) e conseguentemente come IDE PyCharm della suite di JetBrains . Come servizio di hosting è stato scelto GitHub.

1.2 Librerie

- Sklearn
  Scikit-learn (ex scikits.learn) è una libreria open source di apprendimento automatico per il linguaggio di programmazione Python. Contiene algoritmi di classificazione, regressione e clustering (raggruppamento) e macchine a vettori di supporto, regressione logistica, classificatore bayesiano, k-mean e DBSCAN, ed è progettato per operare con le librerie NumPy e SciPy.
- Pandas

Nella programmazione per computer, Pandas è una libreria software scritta per il linguaggio di programmazione Python per la manipolazione e l'analisi dei dati. In particolare, offre strutture dati e operazioni per manipolare tabelle numeriche e serie temporali.

- MathPlotLib
  Matplotlib è una libreria per la creazione di grafici per il linguaggio di programmazione Python e la libreria matematica NumPy. Fornisce API orientate agli oggetti che permettono di inserire grafici all'interno di applicativi usando toolkit GUI generici, come WxPython, Qt o GTK.

2 Preprocessing per il clustering

Nella fase di preprocessing del nostro dataset (spotify_features.csv) abbiamo:

- Creato un indice utilizzando le features 'track_name' e 'artist_name'

- Creato una prima tabella chiamata *attributes,* partendo dal dataset iniziale, rimuovendo 'track_id', 'track_name', 'track_id', 'time_signature', 'track_name', 'artist_name', 'key'

- Creato una seconda tabella chiamata *genres* dove per ogni tipologia abbiamo aggiunto una feature di tipo binario così che ogni canzone abbia 1 al proprio genere. Di seguito è riportata la lista dei generi:

- 'genre_A Capella', 'genre_Alternative', 'genre_Anime', 'genre_A Capella', 'genre_Alternative', 'genre_Anime', 'genre_Blues', "genre_Children's Music", "genre_Children's Music", 'genre_Classical', 'genre_Comedy', 'genre_Country', 'genre_Dance', 'genre_Electronic', 'genre_Folk', 'genre_Hip-Hop', 'genre_Indie', 'genre_Jazz', 'genre_Movie', 'genre_Opera', 'genre_Pop', 'genre_R&B', 'genre_Rap', 'genre_Reggae', 'genre_Reggaeton', 'genre_Rock', 'genre_Ska', 'genre_Soul', 'genre_Soundtrack', 'genre_World'

- Unito le due tabelle *attributes* e *genres* grazie all'indice in un'unica tabella chiamata *songs*

- Eliminato eventuali duplicati

La tabella *songs* si presenta in questo modo:

	track_name	artist_name	genre_A Capella	genre_Alternative	genre_Anime	genre_Blues	genre_Children's Music	genre_Children's Music	genre_Classical	genre_Comedy	genre_Country	genre_Dance	genre_Electronic	genre_Folk	genre_Hip-Hop	genre_Indie	genre_Jazz	genre_Movie	genre_Opera	genre_Pop	genre_R&B	genre_Rap	genre_Reggae	genre_Reggaeton	genre_Rock	genre_Ska	genre_Soul	genre_Soundtrack	genre_World	popularity
0	" La Traviata " : Amami Alfredo (Act II) - Dig...	Maria Callas	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	-0.556765
1	"1点"	Yuki Hayashi	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-0.996569
2	"42" - From SR3MM	Rae Sremmurd	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0.652697
3	"45"	The Gaslight Anthem	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0.377819
4	"6人で(強い方が強い)"	Yuki Hayashi	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1.326422

	acousticness	danceability	duration_ms	energy	instrumentalness	\
0	1.760139	-1.025627	-0.867606	-1.123371	-0.396017	
1	-0.968411	-1.639826	-0.648438	0.725142	1.214458	
2	-1.031128	2.228546	0.024363	-0.030205	-0.489819	
3	-1.036918	-1.289625	-0.274344	1.514651	-0.489819	
4	1.698126	-2.038516	-0.539135	-1.692728	2.495969	

	liveness	loudness	mode	speechiness	tempo	valence
0	0.393352	-0.377133	-0.730526	-0.419178	-1.021740	-1.597743
1	-0.610319	-0.537515	1.368876	-0.398695	-0.924746	-1.428170
2	-0.539709	0.471123	-0.730526	0.022828	0.400417	-0.503401
3	0.312655	0.810392	-0.730526	-0.101149	1.954812	-0.122726
4	-0.433794	-1.188544	-0.730526	-0.462839	0.595796	-1.571980

3 Clustering

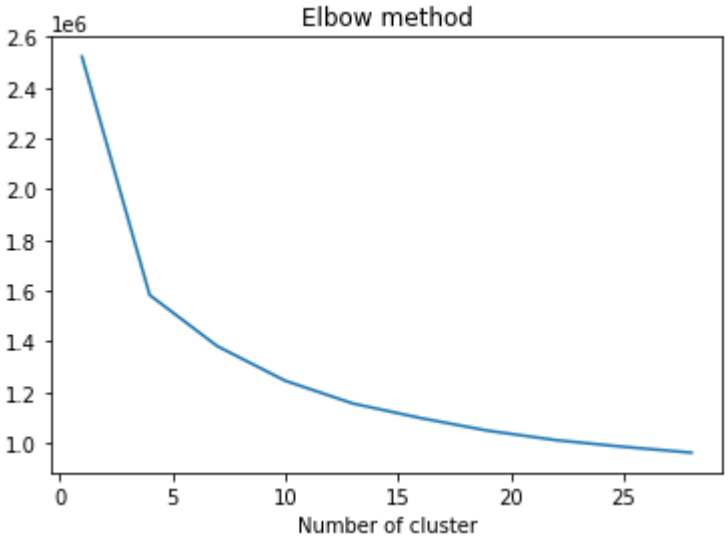
Il clustering consiste in un insieme di metodi per raggruppare oggetti in classi omogenee. Un cluster è un insieme di oggetti che presentano tra loro delle similarità, ma che, per contro, presentano dissimilarità con oggetti in altri cluster. L'input di un algoritmo di clustering è costituito da un campione di elementi, mentre l'output è dato da un certo numero di cluster in cui gli elementi del campione sono suddivisi in base a una misura di similarità.

3.1 K-Means

Nel nostro progetto abbiamo applicato l'algoritmo K-Means. Quest'ultimo ha lo scopo di suddividere un insieme di oggetti in k gruppi sulla base dei loro attributi. L'algoritmo segue una procedura iterativa: inizialmente crea k partizioni e assegna i punti d'ingresso a ogni partizione o casualmente o usando alcune informazioni euristiche; quindi, calcola il centroide di ogni gruppo; costruisce in seguito una nuova partizione associando ogni punto d'ingresso al gruppo il cui centroide è più vicino ad esso; infine vengono ricalcolati i centroidi per i nuovi gruppi e così via, finché l'algoritmo non converge.

3.2 Elbow Method

Applicando l'Elbow Method," Metodo del Gomito", abbiamo scoperto il numero di cluster più adatto per il dataset. L'Elbow Method è stato scelto poiché è un modo totalmente oggettivo per determinare il numero di cluster.



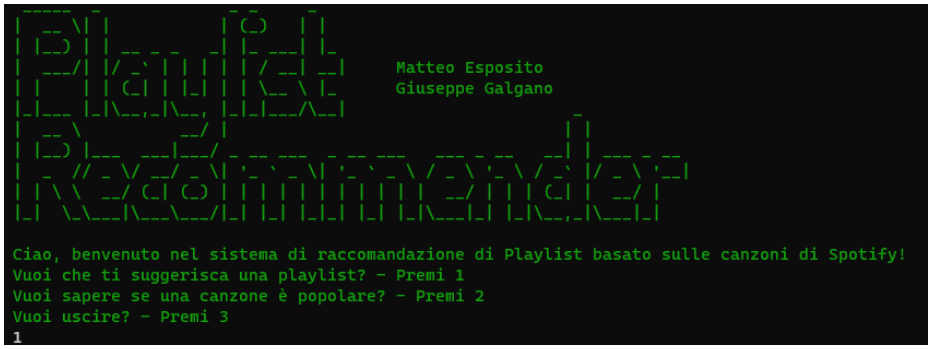
Mettendo su grafico i valori di K (asse orizzontale) e i valori della somma delle distanze al quadrato (asse verticale), si ottiene un grafico simile a quello in figura. Questo grafico deve essere letto da destra verso sinistra. Si deve trovare il punto in cui la curva tende a salire in modo più consistente.
Il numero ottimale di cluster è quello in cui è posizionato il gomito.
Abbiamo scelto di utilizzare la divisione in cinque cluster per avere una scelta più etereogenea. L'utilizzo del clustering assieme al" classificatore" ci ha permesso di ottenere il risultato migliore nel Recommender che vedremo di seguito.

4 Recommender System

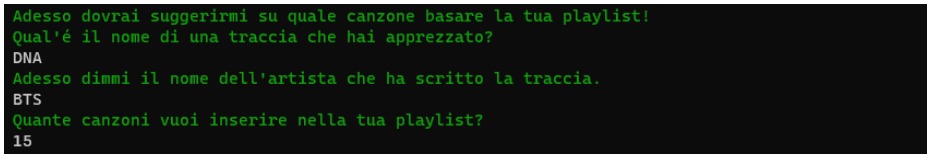
Una volta effettuato il cluster sul dataset si è proceduto ad applicare la raccomandazione basata sulla similarità dei generi. In particolare, si è scelto di utilizzare la similarità del coseno, metrica attraverso la quale è misurato il coseno dell'angolo tra due vettori proiettati in uno spazio multi-dimensionale. Più piccolo è l'angolo più alta sarà la similarità.
Il recommender per consigliare la playlist ha bisogno di chiedere all'utente:

- Nome della canzone
- Nome dell'artista
- Numero di canzoni da inserire nella playlist

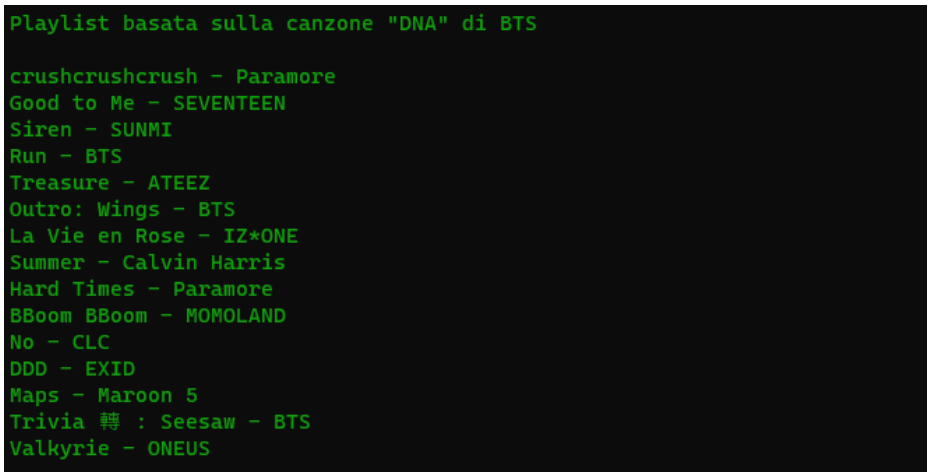
4.1 Guida all'uso



Menù



Domande poste dal sistema per la raccomandazione



Esempio di playlist creata a partire dalle domande poste dal sistema

5 Preprocessing per la classificazione

Per l'attività di classificazione abbiamo apportato le seguenti modifiche al dataset:

- Per la feature key convertiamo le 12 chiavi in numeri utilizzando l'indice
- Per la feature mode convertiamo le Major in 1 e Minor in 0
- Per la feature "time_signature" convertiamo i battiti in numeri, utilizzando l'indice
- Rendiamo la feature "popularity" binaria, una canzone è popolare se ha uno score maggiore o uguale a 75. Non è popolare altrimenti

	genre	artist_name	track_name	track_id	popularity	acousticness	danceability	duration_ms	energy	instrumentalness	key	liveness	loudness	mode	speechiness	tempo	time_signature	valence
0	Movie	Henri Salvador	C'est beau de faire un Show	0BRjO6ga9RKCKjfDqeFgWV	0	0.611	0.389	99373	0.910	0.000	0	0.3460	-1.828	1	0.0525	166.969	0	0.814
1	Movie	Martin & les fées	Perdu d'avance (par Gad Elmaleh)	0BjC1NfoEOOusryehmNudP	0	0.246	0.590	137373	0.737	0.000	1	0.1510	-5.559	0	0.0868	174.003	0	0.816
2	Movie	Joseph Williams	Don't Let Me Be Lonely Tonight	0CoSDzoNIKCRs124s9uTVy	0	0.952	0.663	170267	0.131	0.000	2	0.1030	-13.879	0	0.0362	99.488	1	0.368
3	Movie	Henri Salvador	Dis-moi Monsieur Gordon Cooper	0Gc6TVm52BwZD07Ki6tIvf	0	0.703	0.240	152427	0.326	0.000	0	0.0985	-12.178	1	0.0395	171.758	0	0.227
4	Movie	Fabien Nataf	Ouverture	0IuslXpMROHdEPvSl1fTQK	0	0.950	0.331	82625	0.225	0.123	3	0.2020	-21.150	1	0.0456	140.576	0	0.390

6 Classificazione

Uno degli scopi principali del Machine Learning è la classificazione, cioè il problema di indentificare la classe di un nuovo obiettivo sulla base di conoscenza estratta da un training set.

Un sistema che classifica è detto classificatore. I classificatori estraggono dal dataset un modello che utilizzano poi per classificare le nuove istanze. Il processo di classificazione si può dividere in tre fasi: Addestramento, Stima dell'accuratezza e Utilizzo del Modello. Per lo scopo del nostro progetto abbiamo deciso di suddividere i dati in un insieme di training e un insieme di test fissando quest'ultimo al 20%. La variabile target sulla quale effettuare la predizione sarà la "popularity".

I modelli di classificatori utilizzati sono stati:

- Random Forest Classifier
- K-Nearest Neighbors Classifier
- Decision Tree Classifier
- Logistic Regression

6.1 K-Nearest Neighbors

Uno degli algoritmi più conosciuti nel machine learning è il K-Nearest Neighbors (KNN) che, oltre alla sua semplicità, produce buoni risultati in un gran numero di domini. E un algoritmo di apprendimento supervisionato, il cui scopo è quello di predire una nuova istanza conoscendo i data points che sono separati in diverse classi. Un oggetto è classificato in base alla maggioranza dei voti dei suoi k vicini. K è un intero positivo tipicamente non molto grande. Se k=1 allora l'oggetto viene assegnato alla classe del suo vicino. Il suo funzionamento si basa sulla somiglianza delle caratteristiche, nel nostro caso viene calcolata la similarità delle canzoni nel dataset con la canzone inserita dall'utente. In questo modo si ottiene la variabile target da predire che è la" popularity".

6.2 Random Forest

L'RF o Random Forest Classifier è largamente utilizzato per classificazione, regressione e altri task, funziona costruendo una moltitudine di alberi di decisione. Per la classificazione l'output è la classe selezionata dalla maggior parte degli alberi. La foresta generata dall'algoritmo è addestrata attraverso aggregazione di tipo begging o bootstrap, il begging è un meta-algoritmo ensemble che migliora l'accuratezza degli algoritmi di Machine Learning. L'algoritmo stabilisce il risultato sulla base di predizioni dei decision trees. Esso predice prendendo la media dell'output dei vari alberi, aumentando il numero di alberi si aumenta la precisione del risultato. Il Random Forest elimina i limiti dell'algoritmo Decision Tree, infatti, riduce l'overfitting dei dataset e aumenta la precisione.

6.3 Logistic Regression

Questo tipo di modello statistico è spesso utilizzato per la classificazione e l'analytics predittiva. La regressione logistica stima la probabilità del verificarsi di un evento sulla base di uno specifico dataset di variabili indipendenti. Poichè il risultato è una probabilità, la variabile dipendente è vincolata tra 0 e 1. Nella regressione logistica, viene applicata una trasformazione logit sulle probabilità - ossia la probabilità di successo divisa per la probabilità di fallimento. Ciò è anche comunemente noto come probabilità logaritmica, o logaritmo naturale delle probabilità, e questa funzione logistica è rappresentata dalle seguenti formule:

$$Logit(pi) = 1/ (1+ exp(-pi))$$

$$ln(pi/(1-pi)) = Beta\_0 + Beta\_1*X\_1 + ... + B\_k*K\_k$$

In questa equazione di regressione logistica, logit(pi) è la variabile dipendente o di risposta e x è la variabile indipendente.

6.4 Decision Tree

Nel machine learning un albero di decisione è un modello predittivo, dove ogni nodo interno rappresenta una variabile, un arco verso un nodo figlio rappresenta un possibile valore per quella proprietà e una foglia il valore predetto per la variabile obiettivo a partire dai valori delle altre proprietà, che nell'albero è rappresentato dal cammino (*path*) dal nodo radice (*root*) al nodo foglia. Normalmente un albero di decisione viene costruito utilizzando tecniche di apprendimento a partire dall'insieme dei dati iniziali (*data set*), il quale può essere diviso in due sottoinsiemi: il *training set* sulla base del quale si crea la struttura dell'albero e il *test set* che viene utilizzato per testare l'accuratezza del modello predittivo così creato. Una sua evoluzione è la tecnica del Random Forest.

6.5 Accuratezza dei classificatori

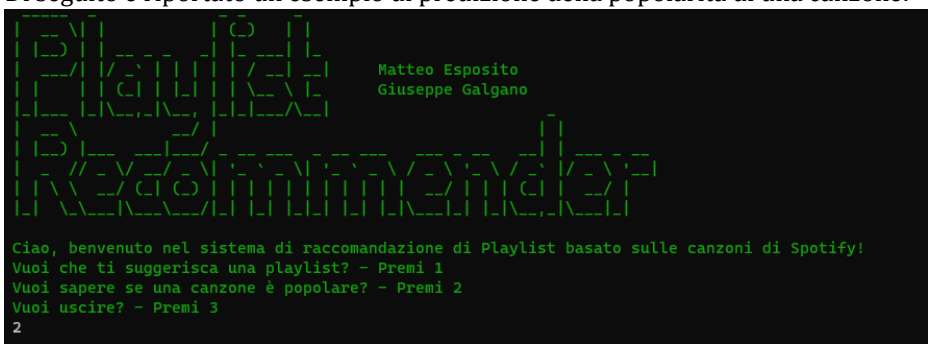
```
Logistic Regression.
Accuracy: 0.98396712858524
AUC: 0.5


Random Forest Classifier.
Accuracy: 0.9940380277151144
AUC: 0.8289000887512975


K Neighbor Classifier.
Accuracy: 0.9814695456010313
AUC: 0.531685831652874


Decision Tree Classifier.
Accuracy: 0.985471049521968
AUC: 0.8261945525547988
```

Risultati dei classificatori

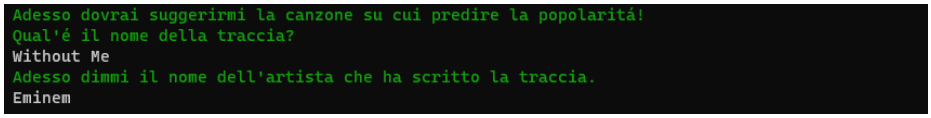
L’esito di questo confronto ci ha portato a scegliere il Random Forest come classificatore per la predizione della popolarità.

6.6 Guida all’uso

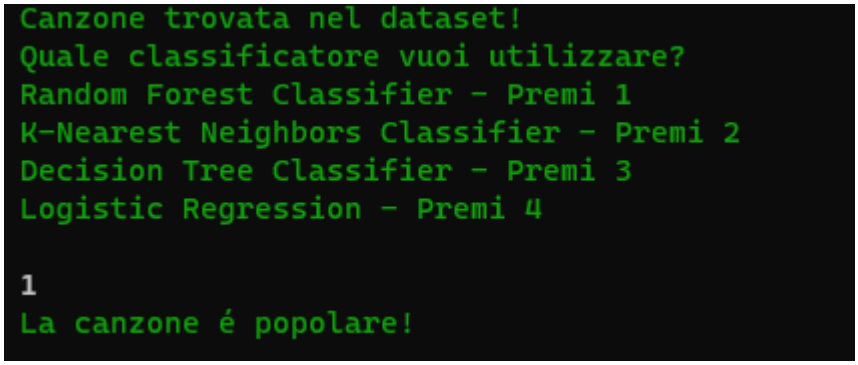
Di seguito è riportato un esempio di predizione della popolarità di una canzone:



Scelta della predizione sulla popolarità



Inserimento del nome del brano e dell’artista



Risultato ottenuto tramite Random Forest