Spotify
Documentazione del caso di studio
AA 2022-2023

Gruppo di lavoro

- Michele Dibisceglia, 735513, m.dibisceglia3@studenti.uniba.it

https://github.com/micheleDibi/ICon-2022-2023

Introduzione

Il progetto è stato sviluppato utilizzando il linguaggio di programmazione Python e si focalizza sull'analisi di brani musicali. I dati relativi a questi brani sono stati raccolti tramite l'API di Spotify, nello specifico attraverso l'utilizzo della libreria Spotipy.

I brani che sono stati raccolti comprendono i "brani preferiti" dell'utente, i quali sono stati estratti da una specifica playlist, personale, per ciascun utente. Inoltre, sono state considerate anche quattro playlist create e costantemente aggiornate da Spotify, che contengono i brani di successo e le tendenze attuali sia in Italia che nel resto del mondo.

Successivamente, sono state effettuate operazioni di pre-processing e i dati sono stati integrati in una base di conoscenza Prolog. Durante questo processo, sono state introdotte nuove clausole per ottenere informazioni aggiuntive attraverso il ragionamento automatico. Tali informazioni sono state successivamente utilizzate per alimentare i modelli di apprendimento automatico.

Infine, è stato affrontato un problema di classificazione dei brani al fine di individuare quelle composizioni musicali il cui insieme di caratteristiche differisce da quello dei brani più ascoltati dall'utente. Questo processo permette di identificare brani che potrebbero risultare interessanti o fuori dall'ordinario, fornendo all'utente l'opportunità di scoprire nuovi stili o artisti musicali.

In particolare:

- Le informazioni raccolte per ciascun brano sono principalmente di natura quantitativa, enumerabile o reale. Questi dati includono caratteristiche come il tempo in battute, l'energia, la “felicità”, l'acustica e altre metriche che descrivono le proprietà musicali dei brani stessi. L'utilizzo di tali dati quantitativi consente un'analisi approfondita e dettagliata del materiale musicale.

- La presenza di una playlist contenente i brani preferiti dell'utente offre una solida base per definire i gusti musicali individuali, comprese le preferenze legate a brani specifici, artisti e generi musicali. Questa informazione è estremamente preziosa poiché semplifica la creazione di nuove caratteristiche che descrivono le preferenze dell'utente in modo più accurato e raffinato. L'analisi delle preferenze musicali dell'utente si basa su tali caratteristiche, consentendo di personalizzare ulteriormente le raccomandazioni e i suggerimenti musicali.

- La diversità di brani presenti all'interno del dataset permette di condurre un'analisi dettagliata dei dati, sfruttando ad esempio tecniche di clustering. Ciò consente di individuare i brani occasionali che sono particolarmente apprezzati dall'utente, nonostante si discostino dalle sue preferenze musicali abituali. Questi brani atipici presenti nella playlist dei brani preferiti forniscono un'opportunità interessante per scoprire nuovi brani o generi musicali che potrebbero suscitare l'interesse dell'utente.

Elenco argomenti di interesse

- **Rappresentazione e ragionamento relazionale**: In questa soluzione, viene impiegato il linguaggio di programmazione Prolog per condurre un ragionamento sofisticato su una base di conoscenza ingegnerizzata, la quale si basa su dati estratti da un'API. Tale approccio consente di inferire nuove informazioni a partire da quelle esistenti, arricchendo così la comprensione e l'analisi del dominio di interesse.

- **Apprendimento supervisionato**: Sono state adottate diverse tecniche di machine learning supervisionato per affrontare compiti di classificazione e generazione automatica di playlist. In particolare, sono stati utilizzati alberi di decisione, random forest, gradient boosting e K-Nearest Neighbors (KNN). Questi algoritmi sono stati addestrati su un ampio set di dati annotati, in modo da imparare dei modelli che consentono di classificare le canzoni in base alle loro caratteristiche o di generare playlist coerenti in modo automatico.

- **Apprendimento non supervisionato**: L'apprendimento non supervisionato è stato impiegato per rilevare anomalie e individuare canzoni che si discostano dai gusti musicali dell'utente. In particolare, è stato utilizzato l'algoritmo del k-means per creare cluster di canzoni simili in base alle loro caratteristiche. Successivamente, è stato possibile identificare quelle canzoni che risultano atipiche o "anomale" rispetto ai brani preferiti dall'utente, offrendo così un'opportunità per scoprire nuovi artisti o generi musicali che potrebbero interessare l'utente.

Raccolta e preparazione dei dati

Il contesto di interesse è incentrato su Spotify, una piattaforma di streaming musicale che permette agli utenti di accedere a brani musicali on demand. La prima fase del processo coinvolge l'autenticazione dell'utente, richiedendo l'inserimento del client_id e del client_secret, necessari per effettuare le richieste alla Web API di Spotify.

Attraverso le richieste alla Web API di Spotify, è possibile ottenere una serie di informazioni riguardanti ciascun brano preferito dell'utente, utilizzando la funzione *current_user_saved_tracks*. Queste informazioni includono:

- "Added_at": La data e l'ora in cui il brano è stato salvato dall'utente.

- "Track":

  - "Album": L'album in cui è contenuto il brano.

  - "Artists": Un elenco di artisti che partecipano al brano. Il primo artista indicato è considerato il "creatore" del brano, mentre gli artisti successivi, se presenti, sono i collaboratori.

  - "Available_markets": Una lista dei paesi in cui il brano è disponibile.

  - "Disc_number": Il numero del disco all'interno di un album (un album può essere composto da più dischi).

  - "Duration_ms": La durata del brano in millisecondi.

- "Explicit": Un valore booleano che indica se il brano contiene o meno riferimenti espliciti.

- "External_ids": Gli identificatori esterni del brano su altre piattaforme di streaming on demand.

- "External_urls": Gli URL esterni del brano su altre piattaforme di streaming on demand.

- "Href": Il link del brano.

- "Id": L'ID del brano su Spotify.

- "Is_playable": Un valore booleano che indica se il brano può essere riprodotto, in accordo con i mercati disponibili.

- "Restrictions": Un elenco di restrizioni specifiche applicate al brano in base al contenuto.

- "Name": Il nome del brano.

- "Popularity": La popolarità del brano, valutata su una scala da 0 a 100, calcolata da un algoritmo che tiene conto del numero di riproduzioni e dell'attualità delle stesse.

- "Preview_url": Un link all'anteprima del brano, che fornisce una preview audio di 30 secondi in formato mp3.

- "Track_number": Il numero del brano all'interno dell'album.

- "Type": Il tipo di contenuto (per i brani, il valore consentito è "track"). Questo viene utilizzato per distinguere i brani dai podcast o dalle playlist, che hanno sezioni distinte.

- "Uri": L'URI del brano su Spotify.

- "Is_local": Una variabile booleana che indica se il brano è locale, cioè un file presente localmente sul dispositivo dell'utente, anziché essere disponibile in streaming on demand.

Per ottenere i brani di ciascuna playlist, viene utilizzata la funzione *playlist* fornendo l'ID della playlist desiderata. In particolare, le informazioni che si ottengono includono:

- "Collaborative": Un valore booleano che indica se ci sono collaboratori coinvolti nella playlist. I collaboratori sono utenti diversi dal proprietario della playlist che possono aggiungere o rimuovere brani dalla stessa.

- "Description": La descrizione associata alla playlist, che fornisce ulteriori informazioni sul suo contenuto o scopo.

- "External_urls": Gli URL esterni che conducono a risorse correlate alla playlist su altre piattaforme.

- "Followers": Il numero di persone che seguono la playlist, indicando il numero di utenti che hanno mostrato apprezzamento o interesse verso la stessa.

- "Href": Il link diretto alla playlist.

- "Id": L'ID della playlist su Spotify, che la identifica univocamente all'interno del sistema.

- "Images": Una lista di link che rappresentano le immagini associate alla playlist. Queste immagini possono includere l'immagine del "profilo" della playlist e l'immagine di copertina che la rappresenta visivamente.

- "Name": Il nome della playlist, che solitamente fornisce un'indicazione sintetica del suo contenuto o tema.

- "Owner": Le informazioni relative al proprietario della playlist, che comprendono dettagli sul possessore come il suo nome o l'ID.

- "Public": Un valore booleano che indica se la playlist è pubblica o meno. Se è pubblica, la playlist è visibile a tutti gli utenti di Spotify; altrimenti, può essere accessibile solo al proprietario e ai suoi eventuali collaboratori.

- "Snapshot_id": Un identificatore che rappresenta la versione specifica della playlist. Viene utilizzato per tracciare eventuali modifiche o aggiornamenti successivi.

- "Tracks": Un elenco di brani inclusi nella playlist. Per una descrizione dettagliata delle informazioni disponibili per ciascun brano, si fa riferimento alla sezione precedente.

- "Type": Il tipo di contenuto della playlist, con il valore consentito per le playlist che è "playlist". Questo attributo viene utilizzato per distinguere le playlist dai podcast o dai brani, che hanno sezioni separate.

- "Uri": L'URI della playlist su Spotify, che fornisce un identificatore univoco per accedere direttamente alla playlist.

Le playlist di Spotify da cui sono prelevati i brani sono:

- Top 50 italia: elenco dei brani più ascoltati in questo momento in Italia. La playlist viene aggiornata quotidianamente.
- Hot Hits: elenco dei brani più ascoltati in tutto il mondo. La playlist viene aggiornata ogni domenica (alle 3 di notte)
- New Music Friday: nuova musica mondiale raccolta settimanalmente. La playlist viene aggiornata ogni venerdì (alle 3 di notte)
- New Music Friday Italia: nuova musica italiana raccolta settimanalmente. La playlist viene aggiornata ogni venerdì (alle 3 di notte)

Per ottenere ulteriori dati per ciascun brano si è utilizzata la funzione *audio_features* dando in input l'id del brano. In particolare, si ottiene:

- acousticness: Una misura di fiducia da 0,0 a 1,0 che indica se il brano è acustico. 1,0 rappresenta un'elevata fiducia che il brano sia acustico.

- analysis_url: link esterno ad un analisi completa del brano
- danceability: La danceability descrive quanto un brano sia adatto al ballo in base a una combinazione di elementi musicali, tra cui il tempo, la stabilità del ritmo, la forza del battito e la regolarità generale. Un valore di 0,0 è il meno danceability e di 1,0 è il più danceability.
- duration_ms: durata del brano in millisecondi
- energy: L'energia è una misura che va da 0,0 a 1,0 e rappresenta una misura percettiva dell'intensità e dell'attività. In genere, i brani energetici sono veloci, forti e rumorosi. Ad esempio, il death metal ha un'energia elevata, mentre un preludio di Bach ha un punteggio basso su questa scala. Le caratteristiche percettive che contribuiscono a questo attributo includono la gamma dinamica, il volume percepito, il timbro, la velocità di insorgenza e l'entropia generale.
- Id: id spotify del brano
- Instrumentalness: Prevede se un brano non contiene voci. I suoni "ooh" e "aah" sono considerati strumentali in questo contesto. I brani rap o parlati sono chiaramente "vocali". Più il valore di strumentalità è vicino a 1,0, maggiore è la probabilità che il brano non contenga contenuti vocali. I valori superiori a 0,5 sono intesi come tracce strumentali, ma la fiducia è maggiore man mano che il valore si avvicina a 1,0.
- Key: La tonalità in cui si trova la traccia. Gli interi corrispondono alle tonalità utilizzando la notazione standard della Pitch Class. Ad esempio, 0 = C, 1 = C♯/D♭, 2 = D e così via. Se non è stata rilevata alcuna tonalità, il valore è -1.
- Liveness: The pitch in which the track is located. Integers correspond to tones using standard Pitch Class notation. For example, 0 = C, 1 = C♯/D♭, 2 = D, and so on. If no pitch was detected, the value is -1.
- Loudness: Il volume complessivo di una traccia in decibel (dB). I valori di loudness sono mediati sull'intero brano e sono utili per confrontare il volume relativo dei brani. Il loudness è la qualità di un suono che è il principale correlato psicologico della forza fisica (ampiezza). I valori sono tipicamente compresi tra -60 e 0 db.
- Speachness: rileva la presenza di parole parlate in una traccia. Più la registrazione è esclusivamente di tipo parlato (ad esempio, talk show, audiolibri, poesie), più il valore dell'attributo è vicino a 1,0. I valori superiori a 0,66 descrivono tracce che probabilmente sono composte interamente da parole parlate. I valori compresi tra 0,33 e 0,66 descrivono tracce che possono contenere sia musica che parlato, in sezioni o stratificati, compresi i casi di musica rap. I valori inferiori a 0,33 rappresentano molto probabilmente musica e altre tracce non simili al parlato.
- Tempo: Il tempo complessivo stimato di un brano in battiti al minuto (BPM). Nella terminologia musicale, il tempo è la velocità o il ritmo di un determinato brano e deriva direttamente dalla durata media dei battiti.
- time_signature: Una firma temporale stimata. La firma temporale (metro) è una convenzione di notazione per specificare quante battute ci sono in ogni battuta

(o misura). La firma temporale va da 3 a 7, indicando firme temporali da "3/4" a "7/4".
- track_href: link del brano
- type: tipo di contenuto (il cui unico valore ammesso per i brani è "track") (viene utilizzato per distinguerlo dai podcast o dalle playlist che hanno una sezione a parte)
- uri: uri spotify del brano
- valence: Una misura da 0,0 a 1,0 che descrive la positività musicale trasmessa da un brano. I brani con alta valenza suonano più positivi (ad esempio, felici, allegri, euforici), mentre quelli con bassa valenza suonano più negativi (ad esempio, tristi, depressi, arrabbiati).

Preprocessing dei dati

Per prima cosa, è stato effettuato un processo di eliminazione dei brani duplicati presenti in più playlist. Inoltre, sono stati rimossi dalla playlist i brani che erano già presenti tra i brani preferiti dell'utente. Questo assicura una selezione unica e personalizzata di brani per l'utente.

Purtroppo, soprattutto per i brani appena usciti, non è presente un genere musicale. Caratteristica di cui si tiene conto per quei brani che la hanno al fine di determinare i generi più ascoltati dall'utente.

Inoltre, è stata fatta una distinzione tra il creatore del brano e i collaboratori che hanno partecipato alla sua realizzazione. Questo permette di riconoscere il contributo di artisti diversi all'interno di un brano.

Le informazioni mantenute per ciascun brano includono:

- ID: l'identificatore del brano su Spotify.

- Nome del brano: il titolo del brano.

- Nome dell'album: il nome dell'album a cui il brano appartiene. Implicitamente, fornisce anche la data di uscita del brano e dell'album stesso.

- Artista: l'artista principale responsabile del brano.

- Collaboratori: un elenco di artisti che hanno contribuito al brano in qualità di collaboratori.

- Generi: un elenco di generi musicali associati al brano. Spesso i generi musicali correlati non differiscono significativamente tra loro.

- Popolarità: indica la popolarità del brano, rappresentando la sua diffusione e gradimento tra il pubblico.

- Tempo: La tonalità in cui si trova la traccia.

- BPM: rappresenta il numero di battiti per minuto del brano. I generi musicali possono avere range specifici di BPM, come ad esempio l'hip hop e il funk che

vanno dai 70 ai 110 BPM, il reggaeton che va dagli 80 ai 110 BPM, la musica disco che va dai 110 ai 140 BPM, e così via. [1]

- Energy: rappresenta l'intensità e l'energia del brano.

- Danceability: indica la facilità con cui il brano è ballabile.

- Happiness: rappresenta il livello di felicità o gioia trasmesso dal brano.

- Loudness: indica il volume generale del brano.

- Acousticness: rappresenta la presenza di elementi acustici nel brano rispetto a quelli elettronici o sintetizzati.

- Instrumentalness: indica la presenza di parti strumentali nel brano rispetto a parti vocali.

- Liveness: rappresenta il livello di percezione della presenza di un'esibizione dal vivo nel brano.

- Mode: indica la modalità tonale del brano (maggiore o minore).

- Speechiness: rappresenta il livello di presenza di parti vocali parlate nel brano.

- Time Signature: indica la firma temporale del brano, ovvero il numero di battute contenute in ogni misura.

- Valence: rappresenta il livello di positività o negatività emotiva trasmesso dal brano.

Si fa notare che i range di battiti per minuto indicati per i generi musicali sono forniti a titolo di esempio e possono variare leggermente a seconda delle fonti e delle preferenze degli artisti e dei produttori.

Al fine di conferire un'importanza maggiore alle preferenze dell'utente, come i generi e gli artisti preferiti, è stata sviluppata una solida base di conoscenze. Tale base di conoscenze ha lo scopo di:

1. Consentire il ragionamento automatico, sfruttando la struttura degli individui e delle relazioni, al fine di inferire nuova conoscenza. Attraverso questo approccio, è possibile ottenere nuove informazioni a partire dai dati esistenti, ampliando così la comprensione dei gusti e delle preferenze dell'utente.

2. Ingegnerizzare nuove caratteristiche booleane che rivestono un ruolo fondamentale nell'ambito dell'apprendimento automatico. Queste nuove caratteristiche consentono di rappresentare in modo efficace e sintetico aspetti rilevanti delle preferenze dell'utente, facilitando così i compiti di apprendimento e di generazione di raccomandazioni personalizzate.

Attraverso l'applicazione di questa conoscenza di base, è possibile ottenere vantaggi significativi nel processo di analisi e comprensione dei dati, consentendo una migliore

modellazione del profilo musicale dell'utente e offrendo un'esperienza di raccomandazione più precisa e personalizzata.

Creazione e Integrazione Knowledge Base

Come detto precedentemente, è stata creata una base di conoscenza utilizzando il linguaggio di programmazione logica Prolog, ed interfacciandosi in Python ad essa mediante la libreria pyswip. Con la creazione della base di conoscenza e popolamento di questa tramite fatti provenienti direttamente dalla WEB API di Spotify, la Knowledge Base è stata sfruttata per ingegnerizzare features per l'apprendimento supervisionato.

Regole

```
appartieneAllArtista(Canzone, Artista) :- appartieneA(Canzone, Album), haCreato(Artisa, Album).

attributo_comune(X, Y, Attributo) :- attributo(X, Attributo, Valore), attributo(Y, Attributo, Valore)

numero_attributi_comuni(X, Y, NumeroAttributi) :- setof(Attributo, attributo_comune(X, Y, Attributo), AttributiComuni), length(AttributiComuni, NumeroAttributi).

hannoGenereComune(X, Y) :- genere(X, Genere), genere(Y, Genere), X \= Y.

simili(X, Y) :- numero_attributi_comuni(X, Y, NumeroAttributi), NumeroAttributi > 7, hannoGenereComune(X, Y)

gradisceCanzoneGenere(A, Canzone) :- haAscoltatoFrequentemente(A, GenereX),  genere(Canzone, GenereX)

gradisceCanzoneArtista(A, Canzone) :- haApprezzatoParticolarmente(A, ArtistaY), haCreato(ArtistaY, Album), appartieneA(Canzone, Album)

coinvolto(ArtistaX, AlbumW) :- haCollaborato(ArtistaX, ArtistaY, CanzoneZ), haCreato(ArtistaY, AlbumW), appartieneA(CanzoneZ, AlbumZ), dataUscita(AlbumZ, DataUscitaZ), dataUscita(AlbumW, DataUscitaW), DataUscitaW @> DataUscitaZ
```

- La regola `appartieneAllArtista(Canzone, Artista)` afferma che una `Canzone` appartiene a un determinato `Artista` se la `Canzone` appartiene a un `Album` creato da quell'`Artista`.

- La regola `attributo_comune(X, Y, Attributo)` afferma che `X` e `Y` hanno un `Attributo` in comune se entrambi hanno lo stesso valore per tale attributo.

- La regola `numero_attributi_comuni(X, Y, NumeroAttributi)` determina il numero di attributi comuni tra `X` e `Y`, calcolando gli attributi in comune tra di loro e restituendo la loro lunghezza.

- La regola `hannoGenereComune(X, Y)` afferma che `X` e `Y` hanno un genere musicale in comune se entrambi sono associati allo stesso genere.

- La regola `simili(X, Y)` stabilisce che `X` e `Y` sono brani musicali simili se hanno più di quattro attributi in comune e condividono almeno un genere musicale.

- Le regole `gradisceCanzoneGenere(A, Canzone)` e `gradisceCanzoneArtista(A, Canzone)` sono asserzioni dinamiche che rappresentano le preferenze di un utente `A`. La prima regola indica che l'utente gradisce una `Canzone` se ha ascoltato frequentemente un genere musicale `GenereX` a essa associato. La seconda regola afferma che l'utente gradisce una `Canzone` se ha apprezzato particolarmente un `ArtistaY` che ha creato l'`Album` a cui appartiene la `Canzone`.

- La regola `coinvolto(ArtistaX, AlbumW)` stabilisce che un `ArtistaX` è coinvolto in un `AlbumW` se ha collaborato con un altro `ArtistaY` nella creazione di una `CanzoneZ`, la quale appartiene a un `AlbumZ` uscito in una data precedente a quella dell'`AlbumW`.

Fatti

I fatti rappresentano informazioni specifiche su brani musicali e sono inseriti nella base di conoscenza Prolog utilizzando le seguenti asserzioni:

- Il fatto `canzone(NomeCanzone)` indica l'esistenza di una `Canzone` con un determinato `NomeCanzone`.

- Il fatto `album(NomeAlbum)` rappresenta l'esistenza di un `Album` con un determinato `NomeAlbum`.

- Il fatto `dataUscita(NomeAlbum, DataUscita)` indica la data di uscita di un `Album` specificato.

- Il fatto `artista(NomeArtista)` rappresenta l'esistenza di un `Artista` con un determinato `NomeArtista`.

- Il fatto `haCreato(NomeArtista, NomeAlbum)` afferma che un certo `Artista` ha creato un determinato `Album`.

- Il fatto `appartieneA(NomeCanzone, NomeAlbum)` indica che una certa `Canzone` appartiene a un determinato `Album`.

Inoltre, sono presenti fatti che rappresentano attributi specifici associati alle canzoni, tra cui:

- Il fatto `genere(NomeCanzone, Genere)` indica che una determinata `Canzone` è associata a un determinato `Genere`.

- Il fatto `popolarita(NomeCanzone, Popolarita)` rappresenta la popolarità di una `Canzone`.

- I fatti `attributo(NomeCanzone, Attributo, Valore)` rappresentano attributi specifici di una `Canzone`, come la tonalità, il BPM, l'energia, la ballabilità, la felicità, il volume e l'acousticness.

Modelli di apprendimento per la classificazione e valutazione

Introduzione

Considerando sia le caratteristiche originali che quelle ingegnerizzate, l'obiettivo dei modelli di apprendimento automatico è quello di prevedere quali brani, all'interno delle quattro playlist, possono essere più coerenti con le preferenze musicali dell'utente, allo scopo di creare playlist personalizzate su misura per lui.

I modelli di apprendimento automatico si basano principalmente su features numeriche e booleane, con queste ultime rappresentate tramite una mappatura su valori binari (0 e 1). Tuttavia, le features categoriche sono molto limitate e la sola categoria che potrebbe essere utilizzata sarebbe quella dei generi musicali, anche se presi singolarmente potrebbero non avere un valore predittivo significativo.

Nello specifico, per il processo di apprendimento sono state impiegate le librerie **Python Scikit Learn**, utilizzando i seguenti algoritmi:

1. **k-Nearest Neighbors (k-NN)**: Questo algoritmo consente di individuare i brani il cui profilo musicale è meno distante da quelli preferiti dall'utente. Si basa sul principio che brani simili tendono a condividere caratteristiche simili.

2. **Decision Tree Classifier**: Questo modello di apprendimento automatico sfrutta un albero decisionale per classificare i brani in base alle loro caratteristiche numeriche. È in grado di gestire efficacemente dati numerici e può essere particolarmente robusto anche in presenza di dati rumorosi.

3. **Random Forest**: Questo algoritmo si basa su un insieme di alberi decisionali e sfrutta il principio dell'aggregazione per migliorare la precisione delle previsioni. È particolarmente adatto per gestire caratteristiche numeriche e può gestire grandi quantità di dati in modo efficiente.

4. **Gradient Boosting Classifier**: Questo modello sfrutta una serie di alberi decisionali, ognuno dei quali cerca di correggere gli errori dei modelli precedenti. Attraverso iterazioni successive, il modello si adatta meglio ai dati di addestramento, fornendo previsioni più accurate.

L'utilizzo di tali algoritmi consente di ottenere modelli predittivi robusti e in grado di elaborare le caratteristiche numeriche delle tracce musicali, contribuendo così alla creazione di playlist personalizzate che riflettano le preferenze dell'utente in modo accurato ed efficace.

KNN

Per il problema specifico di prevedere la feature target "PREFERITO" basandosi sulle caratteristiche numeriche e booleane dei profili dei brani musicali, è stato utilizzato il metodo k-Nearest Neighbors (k-NN) come primo algoritmo di apprendimento automatico.

Il k-NN è un algoritmo di classificazione che si basa sull'idea che gli oggetti simili tendono ad essere vicini l'uno all'altro nello spazio delle caratteristiche. Nel contesto musicale, le caratteristiche numeriche e booleane dei profili dei brani musicali sono utilizzate per rappresentare ciascun brano.

Il processo di addestramento del k-NN coinvolge la creazione di un insieme di dati di addestramento contenente i profili dei brani musicali e le relative etichette "PREFERITO". Successivamente, viene definito un insieme di dati di test per valutare le prestazioni del classificatore.

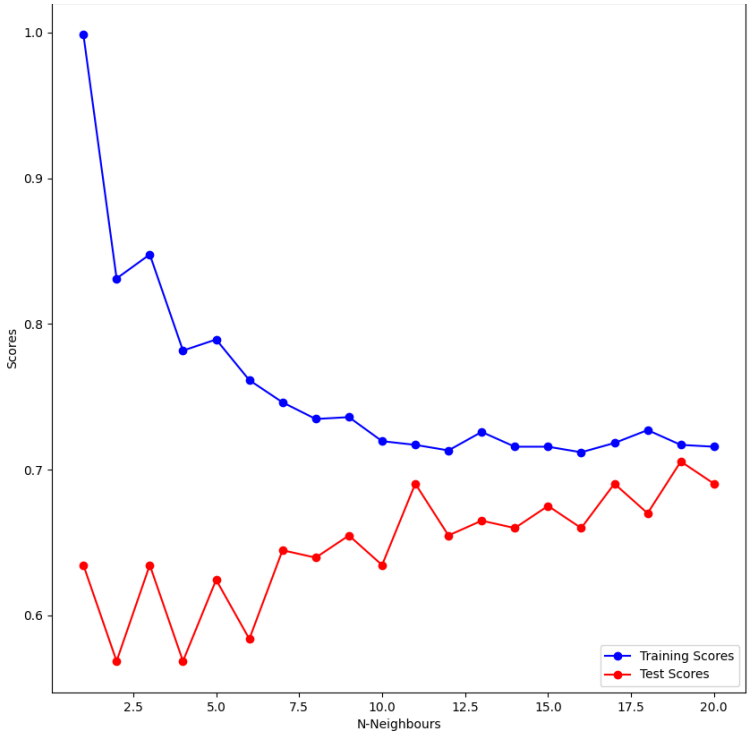
Durante la fase di previsione, per ogni brano del set di dati di test, il k-NN calcola le distanze tra il profilo del brano di test e i profili dei brani di addestramento. I k brani più vicini al brano di test vengono selezionati in base alle distanze calcolate.

Il parametro k nel k-NN rappresenta il numero di vicini più prossimi considerati per prendere una decisione di classificazione. Variazioni nel parametro k possono influenzare le prestazioni del classificatore. Pertanto, sono state effettuate diverse valutazioni delle prestazioni del classificatore utilizzando valori diversi di k.

La valutazione delle prestazioni del classificatore è stata eseguita misurando le metriche di valutazione come l'accuratezza, la precisione, il recall e l'F1-score, al variare del parametro k. Ciò ha permesso di identificare il valore di k che produceva le prestazioni migliori per il problema specifico.

In sintesi, il k-Nearest Neighbors è stato utilizzato per prevedere la feature target "PREFERITO" basandosi sulle caratteristiche numeriche e booleane dei profili dei brani musicali. Sono state eseguite valutazioni delle prestazioni al variare del parametro k per determinare il valore ottimale che massimizzasse le prestazioni del classificatore.

	precision	recall	f1-score	support
0	0.32	0.33	0.33	54
1	0.74	0.73	0.74	143
accuracy			0.62	197
macro avg	0.53	0.53	0.53	197
weighted avg	0.63	0.62	0.63	197



La tabella riporta i valori delle metriche di valutazione, mentre il grafico illustra le prestazioni del modello al variare del parametro k. La linea blu rappresenta l'andamento della curva dell'accuratezza per i dati di addestramento, mentre la linea rossa indica l'andamento dell'accuratezza per i dati di test.

L'accuratezza è una misura che indica la percentuale di previsioni corrette fatte dal modello. Nel grafico, l'asse x rappresenta i diversi valori di k utilizzati, mentre l'asse y rappresenta l'accuratezza del modello.

L'andamento della curva dell'accuratezza per i dati di addestramento (linea blu) mostra come le prestazioni del modello variano al variare di k. Questa curva può fornire informazioni sul comportamento del modello in relazione alla complessità del problema. Ad esempio, se l'accuratezza aumenta all'aumentare di k fino a un certo punto e poi diminuisce, potrebbe indicare che un valore moderato di k è ideale per il modello.

L'andamento dell'accuratezza per i dati di test (linea rossa) indica come il modello si generalizza su nuovi dati, al variare di k. L'obiettivo è ottenere un'accuratezza elevata sia sui dati di addestramento che sui dati di test, indicando che il modello ha appreso in modo efficace e può generalizzare bene su nuovi dati.

L'analisi della curva dell'accuratezza per i dati di addestramento e di test può aiutare a identificare il valore ottimale di k che massimizza le prestazioni del modello senza incorrere in problemi di underfitting o overfitting.

In conclusione, la tabella riporta le metriche di valutazione dei modelli, mentre il grafico fornisce un'indicazione visiva delle prestazioni del modello al variare del parametro k, evidenziando l'andamento dell'accuratezza per i dati di addestramento e di test. Questi strumenti consentono di valutare le prestazioni e selezionare il valore ottimale di k per il k-Nearest Neighbors.

Decision Tree Classifier

Per la valutazione degli alberi di decisione, è stata adottata una procedura di k-fold cross-validation con un valore di k pari a 10. Questo approccio permette di valutare le prestazioni del classificatore utilizzando diverse suddivisioni del set di dati di addestramento e di validazione, riducendo il rischio di ottenere valutazioni sbilanciate o influenzate dalla particolare divisione dei dati.

Durante la fase di addestramento degli alberi di decisione, sono state considerate sempre le stesse caratteristiche per la previsione del target preferito. La scelta di utilizzare le stesse caratteristiche per tutti gli alberi consente di confrontare in modo equo le prestazioni dei modelli al variare di altri parametri, come ad esempio la profondità massima raggiunta dall'albero.

La profondità massima dell'albero è un parametro chiave che influisce sulle prestazioni del modello. Durante la valutazione, sono state misurate le prestazioni del classificatore utilizzando diverse profondità, al fine di individuare il punto ottimale in cui il modello fornisce le migliori prestazioni.

Per quanto riguarda il criterio di selezione utilizzato per la costruzione dell'albero, è stata scelta la misura di "entropy". L'entropy è una misura della disordine o dell'incertezza all'interno di un set di dati e rappresenta la quantità di informazione contenuta nei dati.

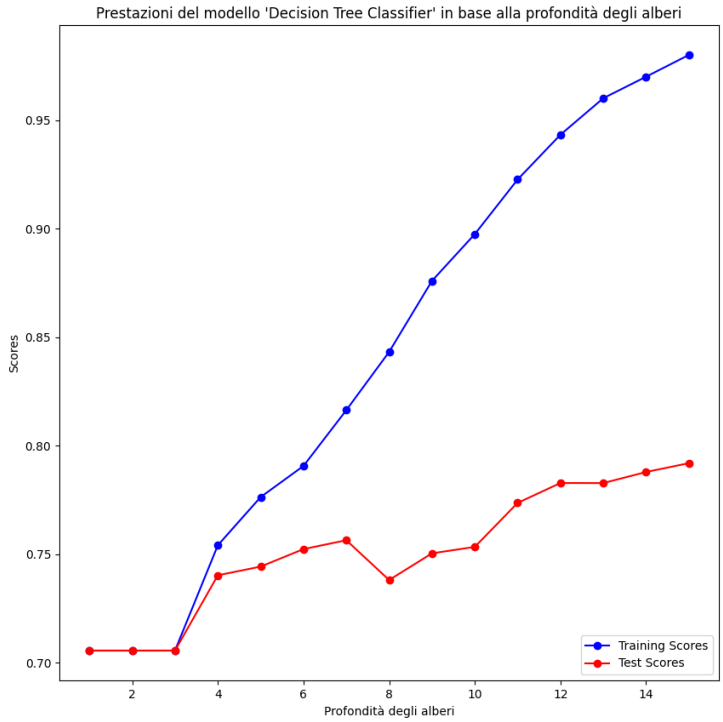
Si precisa che non sono state riscontrate differenze significative in termini di prestazioni utilizzando "entropy" rispetto ad altre misure come "log_loss" o "gini". Questo suggerisce che i risultati ottenuti utilizzando l'entropy come criterio di selezione sono comparabili a quelli ottenuti con altre misure e che l'entropy rappresenta una scelta valida per la costruzione dell'albero di decisione in questo contesto.

Complessivamente, l'approccio utilizzato per valutare gli alberi di decisione ha coinvolto l'addestramento di modelli con diverse profondità massime e l'utilizzo del criterio di selezione "entropy". La valutazione è stata effettuata mediante k-fold cross-validation con un valore di k pari a 10, al fine di ottenere una stima robusta delle prestazioni dei modelli.

Precision: 0.6468440759735099

Recall: 0.6879223725705909

F1 Measure: 0.6580493745163384



Random Forest

Per l'addestramento del modello di apprendimento automatico, è stato utilizzato l'algoritmo di Random Forest. Durante la fase di sperimentazione, è stata esaminata l'accuratezza del modello al variare del numero di alberi appresi e della profondità massima degli alberi.

Nel caso specifico, si è osservato che non vi è stata alcuna significativa variazione dell'accuratezza al variare del numero di alberi appresi, mantenendo una profondità massima di default pari a 5. Questo indica che l'aumento del numero di alberi non ha portato a un miglioramento sostanziale delle prestazioni del modello in termini di accuratezza.

Tuttavia, si è rilevato che la valutazione del modello è stata più significativa al variare della profondità massima degli alberi, mantenendo il valore di default per il numero di alberi appresi. Questo significa che la modifica della profondità massima degli alberi ha avuto un impatto più significativo sull'accuratezza del modello rispetto al numero di alberi utilizzati.

La profondità massima degli alberi rappresenta il numero massimo di divisioni o domande che l'algoritmo può fare durante la costruzione di ciascun albero. Aumentare la profondità massima può consentire al modello di catturare relazioni più complesse nei dati di addestramento, ma potrebbe anche aumentare il rischio di overfitting, in cui il modello si adatta troppo ai dati di addestramento e non generalizza bene sui dati di test.

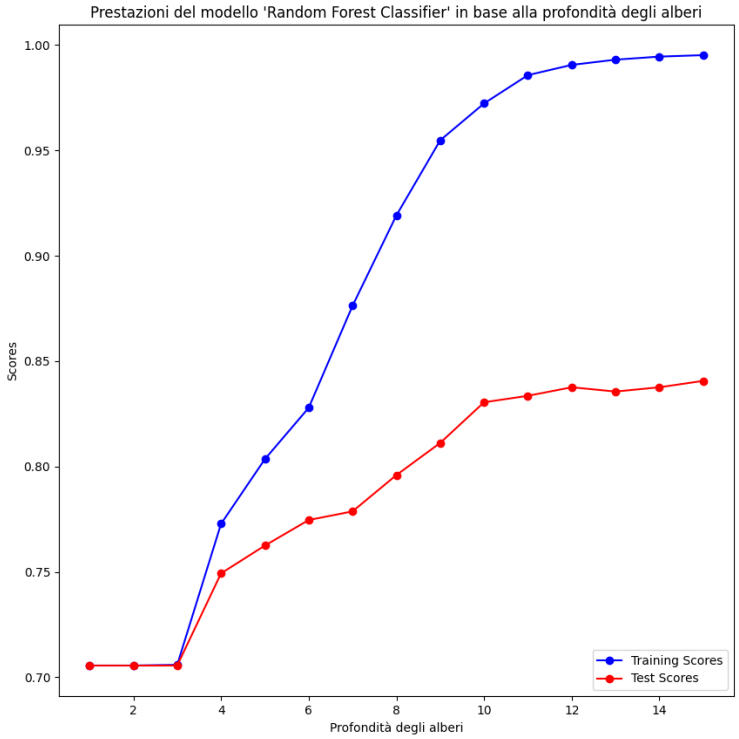
In sintesi, nell'esperimento condotto con l'algoritmo Random Forest, si è rilevato che il numero di alberi appresi non ha avuto un impatto significativo sull'accuratezza del modello, mentre la variazione della profondità massima degli alberi ha avuto un effetto più rilevante. Queste osservazioni indicano l'importanza di valutare attentamente i parametri degli algoritmi di apprendimento automatico per ottenere i migliori risultati in termini di prestazioni del modello.

Training and test scores Random Forest Classifier

Precision: 0.693526309983276

Recall: 0.6823302933672145

F1 Measure: 0.6730848970891234



Gradient Boosting Classifier

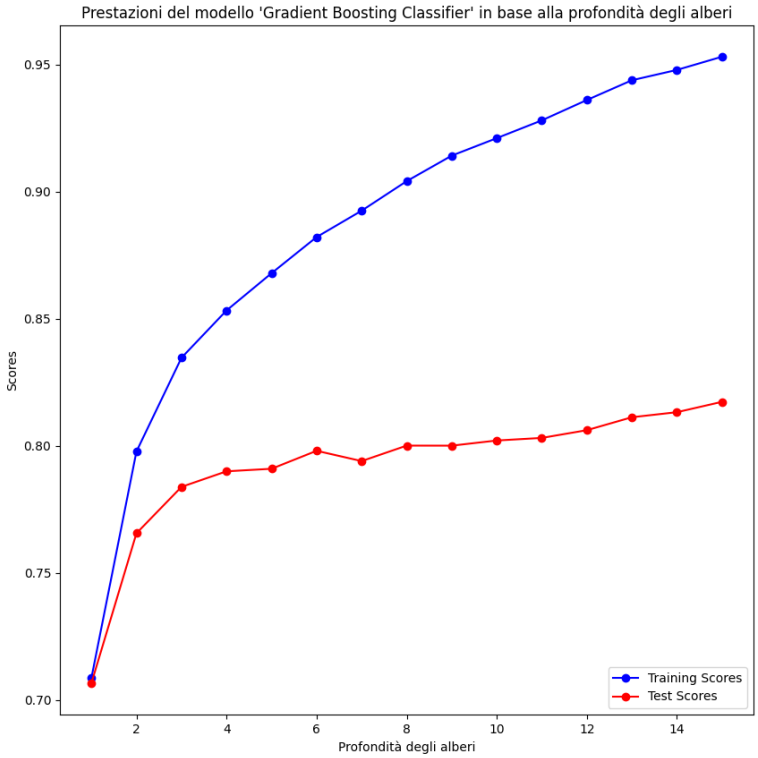
Anche in questo caso si è sperimentato sulla base del numero di alberi utilizzati per l'apprendimento nel modello di boosting.

Training and test scores Gradient Boosting Classifier

Precision: 0.7436083446983514

Recall: 0.7051655592203322

F1 Measure: 0.7104485899853373



Conclusioni

Analizzando i dati forniti, vediamo che hai addestrato tre modelli di classificazione: Decision Tree Classifier, Random Forest Classifier e Gradient Boosting Classifier, insieme a un modello KNN (K-Nearest Neighbors). Ora valuteremo ciascun modello in termini di prestazioni e daremo una valutazione critica, specifica e professionale.

1. Decision Tree Classifier:

- Training Scores: Il modello ha ottenuto un'accuratezza crescente durante l'addestramento, raggiungendo il 97.6% di accuratezza sul set di addestramento finale.

- Test Scores: L'accuratezza sul set di test ha raggiunto il 77.9%.

- Precision: La precisione è pari al 64.2%, il che significa che il modello è in grado di classificare correttamente il 64.2% delle istanze positive.

- Recall: Il recall è del 68.1%, che rappresenta la percentuale di istanze positive correttamente identificate dal modello.

- F1 Measure: L'F1 score, che combina precisione e recall, è pari al 65.5%.

Critica: Il Decision Tree Classifier ha ottenuto risultati accettabili, ma non particolarmente eccezionali. L'accuratezza sul set di test potrebbe essere migliorata per garantire una migliore generalizzazione. Inoltre, la precisione e il recall potrebbero essere migliorati per ridurre gli errori di classificazione.

2. Random Forest Classifier:

- Training Scores: L'accuratezza sul set di addestramento raggiunge il 99.5% dopo l'addestramento finale.

- Test Scores: L'accuratezza sul set di test è del 83.9%.

- Precision: La precisione è del 69.9%, indicando che il 69.9% delle istanze classificate come positive dal modello è effettivamente positivo.

- Recall: Il recall è del 68.5%, che rappresenta la percentuale di istanze positive correttamente individuate.

- F1 Measure: L'F1 score è del 67.0%.

Critica: Il Random Forest Classifier ha ottenuto un'accuratezza generale migliore rispetto al Decision Tree Classifier. Tuttavia, le metriche di precisione, recall e F1 measure non sono significativamente superiori. Ci potrebbe essere una leggera tendenza all'overfitting dato che l'accuratezza sul set di addestramento è molto alta rispetto a quella sul set di test.

3. Gradient Boosting Classifier:

- Training Scores: L'accuratezza sul set di addestramento raggiunge il 95.5% dopo l'addestramento finale.

- Test Scores: L'accuratezza sul set di test è del 82.3%.

- Precision: La precisione è del 74.8%, indicando che il 74.8% delle istanze classificate come positive dal modello è effettivamente positivo.

- Recall: Il recall è del 71.1%, che rappresenta la percentuale di istanze positive correttamente individuate.

- F1 Measure: L'F1 score è del 71.6%.

Critica: Il Gradient Boosting Classifier ha ottenuto risultati simili al Random Forest Classifier in termini di accuratezza e metriche di classificazione. Tuttavia, il modello sembra essere meno incline all'overfitting rispetto al Random Forest Classifier, poiché l'accuratezza sul set di test è ancora elevata.

4. KNN (K-Nearest Neighbors):

- Training Scores: L'accuratezza sul set di addestramento varia tra il 99.9% e l'82.5% durante l'addestramento.

- Test Scores: L'accuratezza sul set di test varia tra il 71.6% e il 59.9%.

- Precision: La precisione media è del 64.8%, il che indica che il 64.8% delle istanze classificate come positive dal modello è effettivamente positivo.

- Recall: Il recall medio è del 68.9%, rappresentando la percentuale di istanze positive correttamente individuate.

- F1 Measure: L'F1 score medio è del 66.1%.

Critica: Il modello KNN mostra una variazione significativa nelle prestazioni durante l'addestramento e sul set di test. Ciò potrebbe essere dovuto alla sensibilità del modello alle scelte dei vicini più prossimi. Inoltre, l'accuratezza sul set di test è inferiore rispetto agli altri modelli considerati.

Confronto tra i modelli:

- Il Random Forest Classifier e il Gradient Boosting Classifier hanno ottenuto risultati simili in termini di accuratezza e metriche di classificazione. Entrambi i modelli sembrano generalizzare meglio rispetto al Decision Tree Classifier.

- Il Random Forest Classifier ha un'accuratezza sul set di test leggermente superiore rispetto al Gradient Boosting Classifier.

- Il modello KNN ha mostrato una precisione, un recall e un F1 score più bassi rispetto agli altri modelli, e le sue prestazioni sono state più variabili.

- Complessivamente, il Random Forest Classifier sembra essere il modello migliore tra quelli considerati, avendo ottenuto un'accuratezza più elevata sul set di test rispetto agli altri modelli, insieme a una buona precisione e recall.

È importante considerare che questa valutazione si basa esclusivamente sui dati forniti e potrebbe essere influenzata da altre considerazioni specifiche del dominio o vincoli del problema. È sempre consigliabile eseguire ulteriori esperimenti e valutazioni per confermare i risultati e ottimizzare ulteriormente i modelli.

Inoltre, ogni modello, crea un file con le canzoni consigliate, rispettivamente:

canzoni_possibili_DecisionTreeClassifier.csv
canzoni_possibili_GradientBoostingClassifier.csv
canzoni_possibili_KNeighborsClassifier.csv
canzoni_possibili_RandomForestClassifier.csv

Clustering

Una sezione fondamentale del progetto è stata dedicata all'individuazione delle anomalie all'interno dei brani preferiti dall'utente. In particolare, si è focalizzata l'attenzione sui brani le cui caratteristiche si discostano in modo significativo dalle preferenze predominanti presenti nella collezione dei brani preferiti.

Al fine di condurre un'analisi esplorativa dei dati, priva di un obiettivo specifico di predizione, si è adottato un modello di apprendimento non supervisionato. Questo

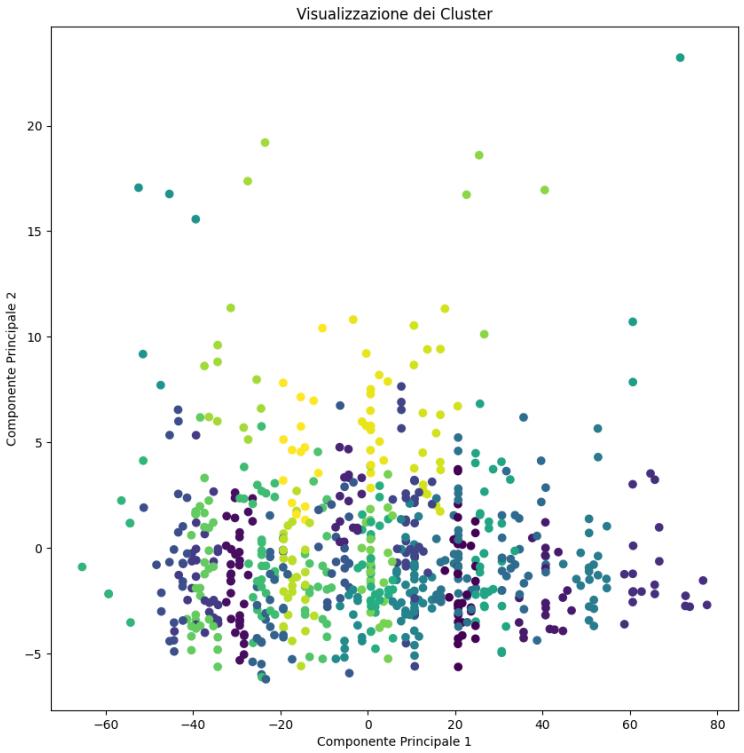
approccio consente di scoprire pattern e strutture nascoste all'interno dei dati, senza la necessità di un output predefinito.

Dopo aver estratto le caratteristiche rilevanti per ciascun brano, è stato impiegato l'algoritmo K-Means per individuare i cluster dei brani. Questo algoritmo, basato sulla distanza tra le caratteristiche dei brani, ha consentito di raggruppare insieme brani simili e di identificare gruppi distinti all'interno della collezione di brani preferiti.

Attraverso questa metodologia di analisi dei dati, è stato possibile identificare brani che si distinguono in modo significativo dal resto delle preferenze dell'utente, fornendo una prospettiva più approfondita sulla diversità e sulla varietà musicale presenti nella sua raccolta. In particolare, sono stati selezionati come anomali quei brani il cui cluster ha la frequenza minore rispetto agli altri.

L'elenco dei brani che vengono definito come anomali viene salvato in un file denominato come *canzoni_anomale.csv*

Il modo ottimale per valutare questo modello di apprendimento non supervisionato è probabilmente quello di effettuare una verifica manuale delle canzoni anomale. Ciò può essere fatto visionando il file "canzoni_anomale.csv" e confrontando le canzoni identificate come "anomale" con le aspettative dell'utente. Questo approccio garantisce che i brani considerati "anomali" effettivamente si discostino dai modelli di ascolto tipici o presentino caratteristiche distintive rispetto alle altre canzoni.



Conclusioni

Per questo progetto, come utente di riferimento sono stato io stesso, avendo un elenco di brani preferiti di quasi 700 brani, vario di ogni genere, da brani storici alle nuove tendenze musicali, dal genere della musica classica alla musica Dubstep, avendo come genere preferito rap / hip hop italiano. Sarebbe curioso testare gli stessi modelli con i brani preferiti di un altro utente.

Come detto precedentemente, non a tutti i brani è associato un genere musicale, nonostante siano presenti tutte le informazioni per poterlo determinare mediante un modello di apprendimento non supervisionato. Potrebbe essere un'idea per possibili implementazioni future.

Riferimenti Bibliografici

[1] https://it.wikipedia.org/wiki/Battiti_per_minuto