**Università degli Studi di Milano-Bicocca**

Corso di laurea Magistrale in Data Science



Report del progetto di Data Management

***Come la popolarità di un artista sui social influenza la sua popolarità musicale***

Report di:

*Merlo Fabrizio 847203*

*Cannatà Alessandra 824434*

**INDICE**

**Abstract**

**1. Introduzione**

**2. Metodologia**

*2.1 Data Acquisition*

*2.2 Data Integration*

*2.3 Data Storage*

**3. Data Quality**

**Appendice**

**Abstract**

Con l’analisi presentata si vuole capire se la popolarità e il numero di ascolti di un artista su Spotify siano correlati alla popolarità dello stesso artista sui social network (in questa analisi, considereremo Instagram come esempio di social network).

Si cerca di sviluppare un’architettura in grado di acquisire ed elaborare un’ampia mole di dati provenienti da fonti diverse, Spotify e Instagram.

Si utilizzerà come DBMS in cui immagazzinare i dati MySQL. I motivi di questa scelta sono molteplici: la sua forma tabellare, la garanzia di avere consistenza e disponibilità dei dati (consistency e availability nel CAP theorem), le proprietà ACID del modello relazionale.

Infine, è stata eseguita l’analisi qualitativa dei dati prendendo in considerazione le dimensioni di affidabilità e correttezza.

**1. Introduzione**

Lo scopo di questo studio è comprendere se la popolarità di un cantante o di una band sui social (Instagram) influenzi la popolarità degli stessi da un punto di vista musicale (quindi, su Spotify).

Di conseguenza, i domini di interesse risultano essere Spotify e Instagram.

Spotify è un servizio di streaming audio. Consente di ascoltare musica e podcast facendo una ricerca per artista, genere, umore, classifica e molto altro. È possibile, inoltre, creare e condividere delle playlist, anche collaborative, usando i propri account personali e associare tracce audio alle storie presenti sul social network Instagram. Dal punto di vista di artisti e speaker, Spotify è un potente mezzo di divulgazione di musica e podcast.

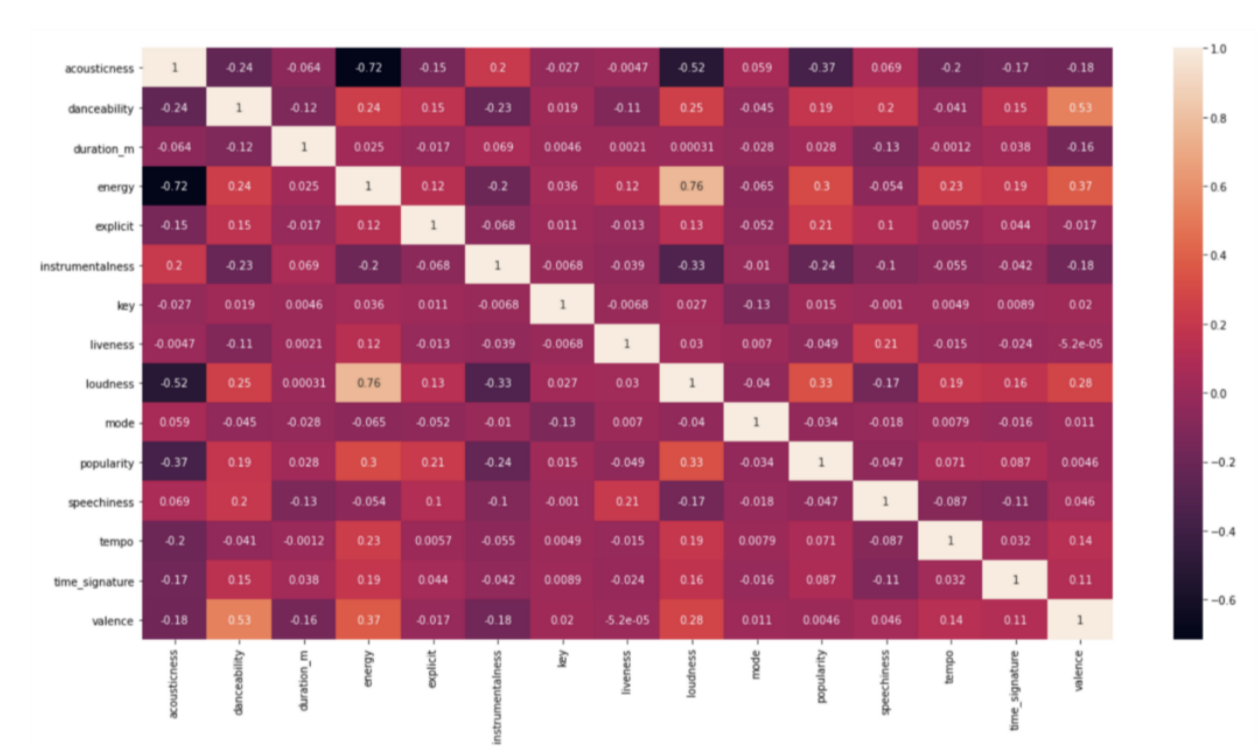
Instagram è nato come social network incentrato sulla condivisione di fotografie, con la possibilità di associare ad esse una descrizione testuale. Attualmente rappresenta anche il principale strumento di comunicazione, divulgazione e trasmissione di contenuti di vario tipo (opinioni, notizie, pubblicità, arte...) di aziende, personaggi noti e profili personali, in forma di foto e video.

Ma qual è la metrica che rappresenta la popolarità?

Per quanto riguarda l’ambito dei social, si è scelta come metrica il numero di follower di Instagram (ossia il numero di persone che segue l’account utente dell’artista e che è interessato agli aggiornamenti dello stesso).

Per quanto riguarda l'ambito musicale, sono state scelte due metriche: l'indice di popolarità calcolato da Spotify, che indica come le canzoni sono percepite dal pubblico che le ascolta (utilizza una scala da 1 a 100, dove maggiore è il valore attribuito a un artista, maggiore è l’apprezzamento che questi riceve dal pubblico) e il numero di ascoltatori mensili dell'artista.

Per quanto riguarda la popolarità calcolata dalla piattaforma musicale, essa si basa sulle caratteristiche delle canzoni dell’artista:



*Fonte dell’immagine:* [*Predicting Popularity on Spotify — When Data Needs Culture More than Culture Needs Data | by Philip Peker | Towards Data Science*](https://towardsdatascience.com/predicting-popularity-on-spotify-when-data-needs-culture-more-than-culture-needs-data-2ed3661f75f1)

Vediamo che i fattori che influenzano maggiormente la popolarità sono l’energy e la loudness, che presentano una correlazione positiva con la popolarità (entrambe pari a circa 0,3). Anche i valori di acousticness e instrumentalness influenzano la popolarità ma presentando una correlazione negativa (rispettivamente pari a -0,37 e -0,24). Tuttavia, le altre grandezze presenti nella tabella non esercitano un impatto molto rilevante sulla metrica popolarità.

Per rispondere alla nostra domanda di ricerca, abbiamo recuperato una mole di dati pari a 18.5MB contenenti le tracce degli artisti più popolari del momento sia classificati geograficamente (per nazione) sia classificati dal punto di vista del genere (pop, rock, indie, hip-hop, jazz…), acquisite tramite scraping. La caratterizzazione delle canzoni, fornita direttamente da Spotify, è stata acquisita tramite richieste alle API di Spotify.

Questo dataset, contenente circa 83.000 record, è stato integrato con ulteriori record raccolti tramite API (il dataset contenente i link delle playlist dalle quali sono stati reperiti questi record è stato scaricato da Kaggle), raggiungendo quindi un numero totale di circa 140.000 righe.

Oltre ai dati acquisiti da Spotify, sono stati immagazzinati tramite scraping il nome dei cantanti utilizzando la EditDistance, cosi da aumentare l’accuratezza della ricerca.

Infine, sui primi 100 cantanti, con il numero di ascolti mensili più alto, sono stati presi i dati provenienti da Instagram, tramite API, contenti il numero di Followers, i Following e il numero di Posts.

La grande mole di dati da scaricare, integrare e processare ha reso necessario l’uso di una metodologia accurata e pertanto verranno fatte in seguito delle considerazioni sulla stessa.

**2. Metodologia**

**2.1 Data Acquisition**

La nostra prima fonte di dati è Spotify, dalla quale abbiamo acquisito dati prima tramite scraping e poi tramite API.

Per trovare i link delle playlist dalle quali acquisire i dati delle canzoni, che sono stati poi elaborati, è stata utilizzata la tecnica dello scraping: è stato creato uno script che prendesse da un csv gli URL che conducono alla pagina contenente tutte le playlist targate Spotify e riguardanti una categoria (ad esempio, la categoria delle playlist settimanali di tutto il mondo o la categoria delle playlist di genere rock, che comprende, quindi, le playlist di tutti i sottogeneri rock: dall’alternative rock al glam rock all’hard rock ecc..). Da questi URL, tramite scraping sono stati recuperati gli URL delle singole playlist. Per svolgere questa operazione sono state impiegate all’incirca 4 ore.

In seguito, a partire da questi URL sono stati acquisiti i dati riguardanti tutte le canzoni interne alle singole playlist (83.000 righe). I dati che acquisiti sono:

* nome artista;
* nome canzone;
* durata della canzone (in minuti);
* popolarità della canzone, popolarità dell’artista;
* numero di follower dell’artista;
* url dell’artista.

La piattaforma mette a disposizione queste informazioni in modo gratuito e le presenta in formato tabellare. Per ottenerle è stato realizzato uno script in Python che inviasse le richieste a Spotify per poter scaricare i dati desiderati. Le richieste sono state realizzate tramite la libreria *Spotipy* di Python.

**2.2 Data Integration**

Per avere a disposizione un numero più elevato di canzoni e artisti tra cui individuare i cantanti più popolari, si è scelto di arrichire il database ottenuto con un dataset[[1]](#footnote-1) proveniente da Kaggle, contenente 1.388 playlist (nome della playlist, descrizione della playlist, numero di tracce, numero di follower, link della playlist). Anche in questo caso, le tracce e i dati relativi a queste playlist sono stati acquisiti tramite le richieste effettuate alle API di Spotify e in seguito sono stati inseriti all’interno del database originario.

Tuttavia, non sono state acquisite tutte le playlist presenti nel dataset Kaggle poiché in molti casi il dataset scaricato presentava al suo interno gli stessi URL ai quali erano già state effettuate precedentemente le richieste. Dunque, per evitare di avere dei duplicati di una stessa playlist, le richieste sono state effettuate solo a una parte del dataset scaricato da Kaggle.

Per svolgere questa operazione sono state impiegate all’incirca 2 ore.

Da questo dataset ottenuto tramite l’integrazione del nostro dataset originario e l’output delle nuove richieste effettuate alle API di Spotify, sono stati estratti, tramite una query, gli artisti con il tasso di popolarità e numero di follower maggiori della media.

Si è creata così un’ulteriore tabella (570 righe) contenente: nome dell’artista (*primary key*), follower, tasso di popolarità, url di spotify dell’artista, numero di apparizioni(calcolato anche esso tramite query). Il nome dell’artista fa da *foreign key* per mettere in comunicazione la tabella appena ottenuta e quella di provenienza.

A questa nuova tabella estratta da quella originaria è stato aggiunto un campo (numero di ascoltatori mensili), i cui valori sono stati acquisiti tramite scraping e il nome di instagram, acquisito anche esso tramite scraping.

Per aumentare l’accuratezza nella ricerca di un cantante su Instagram è stato impiegato il metodo di *edit distance*, il quale, è stato pesato in base alla certificazione del risultato e alla posizione.

Dunque, partendo dalla ricerca su Instagram del nome dell’artista presente nel dataset, sono stati presi in considerazione i primi tre risultati ed è stata quantificata la distanza in termini di similarità tra il nome ricercato da noi e i risultati ottenuti tramite scraping, dando un peso maggiore (pari a 0,4) al primo risultato, poiché generalmente i cantanti più famosi occupano sempre il primo posto nelle ricerche del loro nome. Inoltre, è stato dato un peso ancora più elevato (pari a 0,5) agli account certificati.

La ricerca su Instagram è stata effettuata tramite scraping e il calcolo della edit distance è stato effettuato con la libreria *edit\_distance*.

Una volta ottenuti tutti i risultati, è stato deciso, di selezionare i primi 100 cantanti, quelli con il numero di ascolti mensili maggiori.

Su questi, tramite la libreria *Instagramy* di Python , sono state effettuate delle richieste alle API al fine di ottenere il numero di follower, di following e di post di ogni artista.

**2.3 Data Storage**

I dati sono stati poi salvati su un RDBMS, MySQL. Si è scelto di utilizzare il modello relazionale per diversi motivi: i dati si presentano già strutturati in forma tabellare; l’impiego di un RDBMS assicura la consistenza (consistency) e la disponibilità dei dati (availability), che indicano rispettivamente la possibilità di avere tutti i dati allo stesso momento e la disponibilità dei dati a rispondere a tutte le richieste ricevute.

Inoltre, MySQL, in quanto RDBMS, sfrutta le proprietà ACID:

* Atomicity: l’operazione è atomica, ovvero o avviene per intero o restituisce un errore e il database ritorna alla forma iniziare; ad esempio, se si volessero aggiornare i dati, secondo tale proprietà o si aggiornerebbe tutto il database o non si aggiornerebbe alcun dato.
* Consistency: i nuovi dati inseriti rispecchiano lo schema prestabilito (altrimenti l’operazione di inserimento fallisce).
* Isolation: le singole operazioni non influiscono sulle altre; per fare questo il database costruisce una coda di esecuzione dei processi, così lo stato del database non muta durante l’esecuzione di una singola richiesta.
* Durability: la persistenza del file è garantita anche in caso di crash del sistema.

**3. Data quality**

La metodologia proposta per questo progetto è stata concepita prestando attenzione all’affidabilità della fonte dei dati, alla ridondanza e alla correttezza dei dati stessi.

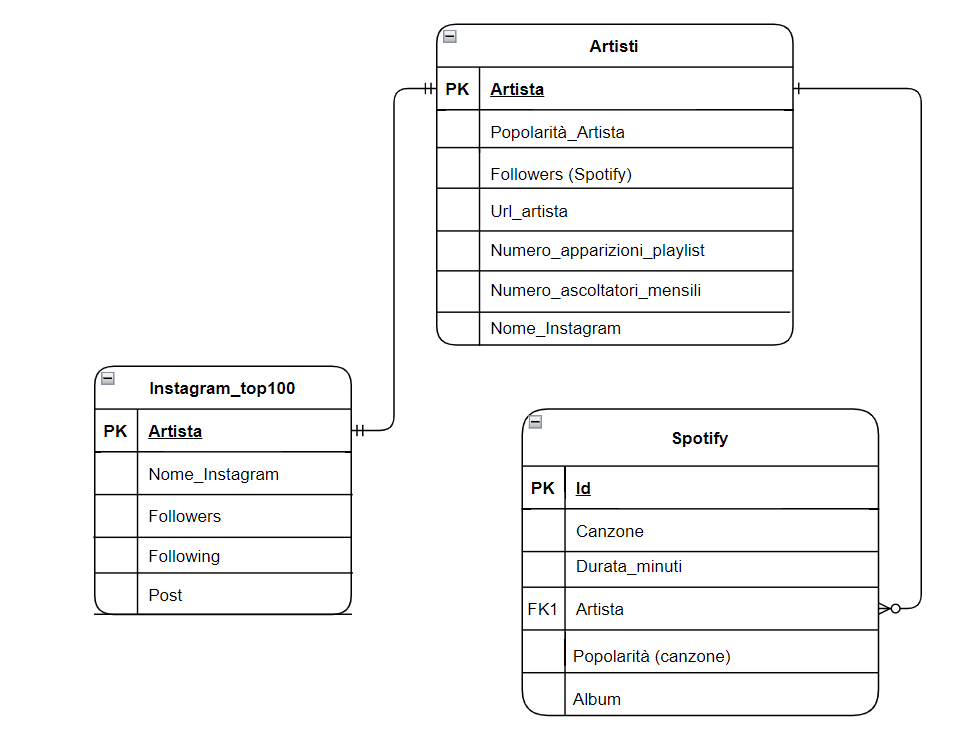
Per garantire l’affidabilità, abbiamo scelto di acquisire solo i dati provenienti dalle playlist targate Spotify. Ciò ha permesso di individuare le canzoni e gli artisti più popolari in maniera oggettiva, mentre prendendo come fonte playlist create dagli utenti avremmo potuto avere a che fare con dati troppo soggettivi e, quindi, poco affidabili.

Per quanto riguarda la correttezza dei dati, essa viene garantita in due modi. In primo luogo, dal fatto che i dati provengono dal sito di Spotify e, di conseguenza, sono certificati. In secondo luogo, dalla metrica scelta per individuare gli account degli artisti più popolari su Instagram: attribuendo agli account risultanti della ricerca diverso peso in base alla posizione e alla certificazione, siamo riusciti a trovare i profili ufficiali degli artisti e, di conseguenza, a ottenere dati corretti.

Infine, è stato gestito anche il problema della ridondanza in quanto, mediante una condizione, vengono aggiunte le playlist che non sono ancora state aggiunte, mentre quelle già inserite in precedenza non vengono considerate.

**Appendice**

Schema del database creato.



1. [Spotify Official Playlists (1388 playlists) | Kaggle](https://www.kaggle.com/mogura20/1388-spotify-official-playlists2021) [↑](#footnote-ref-1)