# The Netflix Dilemma: la percezione delle serie tv tra IMDB e Twitter

Laura Nembrini, Adele Zanfino, Antonio Lombardo, Gabriele Strano
*Università degli Studi di Milano-Bicocca, CdLM Data Science*

05 Giugno 2021

**Sommario**

Un servizio che sta prendendo sempre più piede negli ultimi anni è quello dello streaming on demand. La diffusione di questi servizi è dovuta alla loro facilità di fruizione, in quanto è sufficiente una connessione internet per poter accedere ai propri programmi preferiti tramite qualsiasi dispositivo ed in qualsiasi momento. Tale progetto si propone di analizzare le serie tv presenti in una delle piattaforme di maggior diffusione, ovvero Netflix. L'obiettivo è stato quello di soddisfare due delle tre V dei Big Data, la velocità e la varietà dei dati. Per quanto riguarda la prima, è stata implementata una architettura per la raccolta di dati provenienti da una fonte streaming, Twitter; per quanto riguarda la varietà, invece, sono stati utilizzati dati proventienti da fonti differenti e con formati diversi. Le domande di ricerca che ci siamo posti riguardano principalmente il confronto tra il rating fornito da IMDB e il gradimento ottenuto attravero l'analisi della sentiment dei tweet.

## Indice



## 1 Introduzione

Il mondo dell'intrattenimento ha subito profondi cambiamenti per via della comparsa di numerose piattaforme di streaming. Queste piattaforme sono riuscite inaspettatamente a surclassare la vendita di supporti fisici per via della grande facilità di utilizzo, nonché della vasta scelta che si può avere a portata di click. Tra le numerose piattaforme esistenti, abbiamo scelto di analizzare Netflix, essendo quella più in voga negli ultimi anni. I contenuti di Netflix vengono periodicamente aggiornati (le nuove uscite mensili generano grande attesa tra i fruitori del servizio), generando quindi un catalogo in continuo cambiamento. Un interessante rapporto, rilasciato nel 2018 dalla piattaforma, mostra che il numero di spettatori Netflix è triplicato rispetto

al 2010 mostrando un andamento nettamente in crescita. Esiste un sito, chiamato IMBD, che si occupa di raccogliere, oltre alle informazioni sui vari film e le varie serie tv, anche le recensioni degli utenti, che vanno a comporre un indice di gradimento dei contenuti. Questo studio si propone dunque di utilizzare le informazioni raccolte, compresa un'analisi dei sentiment raccolti sul social network Twitter, per andare a capire quanto il rating fornito da IMDB è veritiero rispetto al giudizio popolare del film o serie tv. Inoltre, si andrà a studiare il genere più apprezzato dagli utenti ed il paese di produzione per capire quali fra questi ha un indice di gradimento maggiore.

# 2 Implementazione dell'architettura

Oltre agli obiettivi già descritti, il progetto si prefissa di soddisfare due delle tre **V** caratteristiche dei Big Data:

- **V**elocità: l'uso di un'applicazione di raccolta dati proveniente da una fonte streaming, per risolvere problemi come la stabilità e durata della connessione per l'immagazzinamento dei dati in real time;
- **V**arietà: l'utilizzo di differenti fonti di dati con diversi formati per la realizzazione di un unico e informativo database. Affrontare, quindi, tutte le problematiche connesse all'integrazione di dati che puntano allo stesso oggetto ma descritto in maniera diversa.

## 2.1 Sorgente dei dati

Per rispondere alle domande di ricerca presentate in precedenza, abbiamo utilizzato dati provenienti da fonti eterogenee. Ci siamo avvalsi di due datasets presenti sulla piattaforma Kaggle. Il primo, in formato csv, riguardante film e serie tv aggiunti sulla piattaforma Netflix dal 2008 ad oggi. È formato da 12 colonne che forniscono informazioni essenziali come titolo, tipologia, data di aggiunta sulla piattaforma, paese di produzione e regista del film. Il dataset contiene, inoltre, 7786 righe. Il secondo dataset, in formato tsv, è relativo ai dati ottenuti dalla piattaforma IMDB, ed è stato utile per arricchire il primo dataset. Composto da 13 colonne e 2000 righe, si sono, infatti, potute aggiungere voci molto utili ai fini della nostra analisi come il genere del film o della serie tv, la durata della serie, il rating IMDB (rating che può essere espresso solo dagli utenti registrati che possono esprimere un indice di gradimento che va da 1 a 10) ed il numero di voti espressi da pubblico e critica per ottenere tale rating. Oltre a questi due datasets, abbiamo poi raccolto molti Tweets relativi ai più famosi prodotti presenti sul catalogo di Netflix.

## 2.2 Streaming da twitter, Apache Kafka

Per raccogliere le opinioni ed i pensieri degli utenti riguardo ai film e le serie considerate, abbiamo eseguito lo streaming dei tweet grazie alla libreria Tweepy di Python, la quale grazie alle API di Twitter permette di "mettersi in ascolto" degli argomenti relativi alle chiavi di ricerca che vengono specificate. Per gestire il flusso continuo di tweet che arrivava abbiamo creato un consumer ed un producer di Apache Kafka, avvalendoci della libreria Kafka-Python. Kafka è una tecnologia middleware che funziona grazie ad un meccanismo a coda, nella quale i dati vengono memorizzati in pile di dati indicizzate. L'architettura di Kafka è composta da due flussi separati: il primo relativo al Producer è quello che scrive il dato nella coda, in questo caso i Tweet. Il secondo, quello del Consumer si occupa di leggere solo i dati relativi agli argomenti desiderati. La caratteristica più importante di questa architettura è quella di rendere la comunicazione tra le componenti asincrona, permettendo così un efficace gestione dell'elevata velocità del flusso dei dati.

### 2.2.1 NiFi

Il producer ed il consumer Kafka sono stati collegati ad Apache NiFi, una piattaforma integrata di logistica dei dati per l'automazione dello spostamento degli stessi tra diversi sistemi attraverso cui è possibile semplificare lo spostamento dei dati tra qualsiasi origine e qualsiasi

destinazione, consentendo inoltre di monitorare i dati in tempo reale. I principali vantaggi di questo sistema sono la velocità, la stabilità, la possibilità di gestire dati non strutturati e in real-time. Questi sono alcuni dei motivi che hanno fatto si che scegliessimo questa piattaforma nel nostro progetto. Apache NiFi utilizza dei processori che consentono di eseguire una grande varietà di operazioni, tra cui ingestion dei dati da sorgenti esterne, trasformazioni del formato dei dati e pubblicazione del contenuto su sistemi differenti. Tra i diversi processori che NiFi mette a disposizione abbiamo deciso di utilizzarne uno di tipo ConsumeKafkaRecord: questo processore interroga Apache Kafka utilizzando l'API KafkaConsumer disponibile con Kafka 2.0. Quando un messaggio viene ricevuto da Kafka verrà in un primo momento deserializzato utilizzando il Record Reader configurato, poi serializzato con il Record Write configurato. Nelle proprietà del nodo abbiamo quindi indicato tra le altre cose, la porta di collegamento (nel nostro caso 9092), il nome del topic e il formato. Per stabilire un collegamento tra i vari processori utilizzati, NiFi mette a disposizione delle entità chiamate Connection. Queste agiscono come buffer intermedi e permettono ai diversi processori di lavorare a diverse frequenze. Abbiamo deciso di configurare due diverse entità di tipo Connection che permettono di collegare il nodo precedentemente descritto sulla base del successo o dell'insuccesso del lettore Kafka. Se non è possibile analizzare un messaggio di Kafka utilizzando il lettore di record configurato, il contenuto del messaggio verrà instradato in un componente di tipo Funnel, che viene utilizzato per combinare i dati di più connessioni in un'unica connessione. In caso di successo, invece, i dati vengono instradati in un processore chiamato PutMongoRecord, che è un nodo in grado di registrare dati perché vengano inseriti in MongoDB. Questo utilizza un lettore di record e uno schema configurati per leggere un set di record in entrata dal processore e quindi inserisce batch di tali record in una raccolta MongoDB configurata, indicata da noi nelle proprietà. Il numero di documenti da inserire di volta in volta è controllato dalla proprietà di configurazione "Inserisci dimensione batch". Questo valore dovrebbe essere impostato su una dimensione ragionevole per garantire che MongoDB non venga sovraccaricato con troppi inserimenti contemporaneamente. Allo stesso modo del nodo ConsumeKafkaRecord, anche il nodo di inserimento sulla piattafroma Mongo è stato collegato a due diversi elementi di tipo Connection. In caso di fallimento, i dati vengono posti nello stesso Funnel utilizzato dal processore Kafka, in caso di successo vengono invece correttamente inseriti in MongoDB. È possibile, inoltre, controllare quanti elementi ci sono in coda attraverso i due nodi di tipo Connection. Nella figura sottostante è possibile vedere il template completo utilizzato.

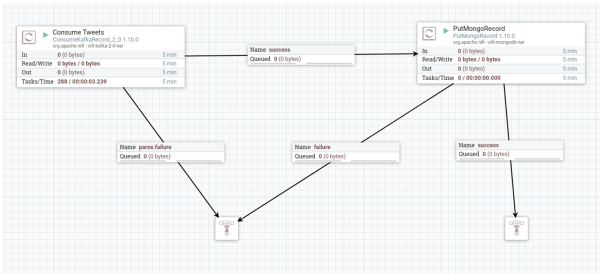


Figura 1: Architettura NiFi

### 2.2.2 Immagazzinamento MongoDB

Per il salvataggio dei dati ci siamo affidati a MongoDB, DBMS non relazionale, orientato ai documenti. A differenza dei database relazionali che hanno una struttura rigida, MongoDB archivia i dati sotto forma di documenti di tipo JSON (BSON chiamati da MongoDB) rendendolo un sistema più flessibile e dinamico, più facile nell'effettuare aggiornamento o modifiche nei dati, garantendo un'integrazione più rapida, veloce e facilmente scalabile. Le transazioni godono delle 4 proprietà ACID complete, indispensabili per l'esecuzione corretta e sicura.

- L'atomicità di una transazione garantisce che i dati siano resi visibili del tutto secondo un approccio "tutto o niente";
- La consistenza di una transazione rappresenta la capacità di rispettare i vincoli referenziali e di integrità dei dati. In caso contrario, l'operazione dovrà essere annullata;

- L'isolamento assicura che le operazioni, eseguite dal DBMS, siano fra loro indipendenti ed isolate in modo che sia possibile una esecuzione parallela;
- La persistenza si ha quando l'effetto delle transazioni eseguite con successo, sia sempre mantenuta nel database anche in caso di guasti o malfunzionamenti.

MongoDB, inoltre, mette a disposizione un servizio di monitoraggio e backup e delle utility command-line, come per esempio mongoimport che permette di importare il file in formato JSON, CSV o TSV creato in precedenza da mongoexport. Si è scelto di utilizzare questo tipo di database poiché, in un contesto lavorativo di gruppo in cui si lavora con più macchine, si è dovuto garantire una struttura che non fosse stata in locale. Inoltre, andando a lavorare con una grande quantità di dati ci si è assicurati che la rottura o un down temporaneo di un nodo non andasse a rovinare o, addirittura, perdere il lavoro svolto. A livello di DB l'uso del modello documentale è stato indispensabile a causa dell'argomentazione trattata poiché non è presente una struttura nei dati. Infatti, andando a valutare argomenti cinematografici, si poteva disporre di un insieme di coppie nome/valore diverse a secondo del tipo di film o serie TV.

### 2.3 Pulizia tweets

La fase successiva a quella della raccolta e dello storage dei tweet riguardanti gli argomenti di nostro interesse, è stata quella della pulizia ed ulteriore selezione di questi ultimi. La grande mole di dati raccolti presentava parecchi tweet identici fra loro, a causa dei numerosissimi retweet. Altra confusione è stata creata dall'acquisizione di tweet che sembravano non riguardare gli argomenti che avevamo impostato come focus delle nostre analisi. Questo è dovuto al fatto che lo streaming effettuato tramite la libreria Tweepy e le API di Twitter effettua la ricerca su ogni campo dei tweet e non si limita soltanto alla voce Text, che è quella di maggior contenuto informativo. Questo comportava che se il nome di un utente o un link contenevano al loro interno una delle stringhe da noi cercate, ci saremmo ritrovati con un tweet poco attinente ai nostri argomenti. Per ovviare a questi due problemi abbiamo, inizialmente rimosso tutti i tweet che presentavano valori identici all'interno della colonna Text e sempre sulla stessa colonna abbiamo poi effettuato la ricerca delle nostre parole chiave, tramite il comando str.contains, in modo da escludere tutti i tweets che pur non riguardando i nostri argomenti erano stati acquisiti da Tweepy.

Ulteriore fase di pulizia è stata necessaria, anche in una fase successiva, in quanto dall'analisi di piccoli campioni di tweet per ognuno degli argomenti, sono emerse ambiguità in merito agli argomenti delle ricerche.

### 2.4 Sentiment analysis

Abbiamo utilizzato la libreria NLTK per analizzare l'opinione degli utenti di twitter rispetto ai prodotti considerati nella nostra ricerca. Una volta completata la fase di pulizia sulla collection contenente i tweet, abbiamo dovuto eseguire una nuova fase di pulizia, stavolta all'interno dei tweets stessi, nello specifico sulla colonna text. Questo passaggio è stato necessario per poter calcolare correttamente degli indicatori per la sentiment analysis. Abbiamo quindi, tramite un'unica funzione, rimosso dal testo dei tweets: le emoticons, la punteggiatura, simboli come RT, i collegamenti ipertestuali ed un corpus di parole che NLTK chiama "Stopwords" cioè quelle parole che fungono da congiunzione ma non aggiungono significato alla frase. Una volta puliti i tweets da queste parti superflue, abbiamo spezzettato la frase in singole parole, tramite la funzione tokenizer di NLTK. Questo perché NLTK garantisce risultati migliori quando analizza singolarmente lo score delle singole parole all'interno di una frase e ne calcola successivamente la media. Arrivati a questo punto avevamo ottenuto per ogni tweet una colonna contenente un dizionario al cui interno vi erano le chiavi: positive, negative, neutral e compound ed i rispettivi valori. Dopo esserci documentati al riguardo, abbiamo scelto di non affidarci al valore della voce compound ma di creare una nuova colonna di score che contenesse il valore più alto tra positive e negative o il

valore neutral quando nessuno dei due prevaleva sull'altro. Per ottenere questa nuova colonna abbiamo utilizzato la funzione np.where di Numpy e successivamente abbiamo trasformato la colonna risultante da array Nunmpy a float, in modo che la colonna potesse essere più agile da utilizzare, contenendo solo valori numerici utili per le successive lavorazioni.

### 2.5 Integrazione

Per l'integrazione dei due database è stato utilizzato in linguaggio Python e le librerie Numpy e Pandas. In particolar mondo la libreria Pandas che fornisce strutture e strumenti per l'analisi dei dati e consente di esplorare i dati in formato Pandas dataframe. Abbiamo deciso di integrare questi due diversi database utilizzando come criterio il titolo dei contenuti. Si è, successivamente, stabilito di analizzare unicamente delle serie televisive, essendo questa tipologia di contenuto la più popolare all'interno della piattaforma Netflix. Come formato, per questa prima integrazione, è stato scelto il "csv"( Comma-Separated Values), in quanto rappresenta uno dei modi più semplici di rappresentare i dati. In una prima fase di pulizia si è stabilito di eliminare le seguenti colonne dal primo database integrato:

- Abbiamo mantenuto una sola colonna contenente il titolo della serie, in quanto si ripeteva in entrambi i database;
- Sono state eliminate anche le colonne che si riferivano alle quattro star principali del prodotto, in quanto nel database di Netflix era presente un campo più completo in cui erano indicati più attori e questo permetteva di fare analisi anche riguardanti personaggi secondari delle serie televisive;
- La colonna Listed_in, in quanto la colonna Genre conteneva la stessa informazione;
- Il link del poster, che per la tipologia di analisi risultava inutile;
- Le colonne Description e Overview, entrambe contenenti piccole descrizioni della trama;
- Release Year, Runtime_of_Series e Date_added, anche queste inutilizzate nelle analisi scelte.

Per quanto riguarda la colonna cast invece, si è deciso di non effettuare un'operazione di split in quanto il numero di attori contenuti non è sempre identico in ogni serie televisiva analizzata e avremmo avuto una presenza elevata di valori NA, e un dataset con numerose colonne inutili.

### 2.6 Seconda integrazione

La seconda fase di integrazione prevedeva invece di integrare la precedente unione dei database di Neflix e IMDB con il database, in formato JSON contenente i tweet che si riferivano alle serie televisive. È stata utilizzata, per questa seconda fase, la libreria Tweepy, già utilizzata nella prima fase per ottenere i tweet. Il database contenente i tweet è stato salvato ed è stato poi prelevato da MongoDB. Il campo id del tweet conteneva, per ogni singolo tweet la chiave con cui era stato salvato. Abbiamo deciso di eliminare questa chiave in quanto non era utile al nostro processo. Inizialmente è stata effettuata una prima pulizia del testo del tweet in modo tale da rimuovere le parti non influenti (come ad esempio la sigla RT, che sta per retweet, i link, gli account menzionati) ed inoltre è stata utilizzata la libreria emoji per eliminare tutte le emoticon presenti nel testo. E' stata applicata poi un ulteriore funzione per eliminare la punteggiatura. Una volta effettuata la pulizia, è stata fatta l'integrazione tra il documento json ricavato dallo stream dei tweet e il csv dato dall'unione tra Netflix e IMDB, in modo tale da arricchire ulteriormente le informazioni legate alle varie serie tv. La scelta è stata quella di contenere tutti i documenti all'interno di un unico documento JSON. Successivamente, abbiamo modificato la struttura del nostro database json, in quanto avevamo un database che conteneva un elemento per ogni tweet, quindi per ogni id_film e title risultavano diverse ripetizioni. Tramite una struttura composta da più cicli for, abbiamo annidato i vari Tweet dentro i film a cui essi si riferivano. In questo modo ab-

biamo ottenuto la struttura definitiva del nostro database.

Abbiamo poi aggiunto un nuovo attributo ad ogni film, chiamato avg_score. Come si intuisce dal nome stesso, per ogni film rappresenta la media del valore score 2 di tutti i Tweets che lo riguardano. In questo modo abbiamo ottenuto un valore di sintesi facile da utilizzare ed utile per effettuare i confronti tra i film, i generi,le nazioni di produzione ecc. Il dataset completo si compone quindi dei seguenti campi: show_id, type, title, director, cast, country, rating, duration, certificate, runtime_of_episodes, genre, IMDB_rating, no_of_votes, tweet, avg_score.

# 3 Query sul database

Per effettuare il confronto tra il valor medio dello score della sentiment analysis e il rating di IMDB abbiamo standardizzato le due colonne, in modo tale da portare i valori ad una stessa scala e non creare distorsioni. Per testare che la trasformazione fosse andata a buon fine abbiamo anche verificato che la media e la deviazione standard di entrambe le colonne fossero uguali, rispettivamente a 0 ed 1. Abbiamo poi aggiunto una colonna che mettesse in mostra le principali discordanze tra i due valori, per verificare quanto l'opinione degli utenti Twitter fosse concordante con quella dei votanti su IMDB. Ci siamo poi concentrati sul genere delle serie tv ed abbiamo visto che anche in questo caso entrambi i valori, tendenzialmente, si muovevano in maniera concordante, il genere preferito è risultato Drama seguito a stretto giro da Adventure e Comedy. Per svolgere questa analisi abbiamo prima tenuto in considerazione i rating uno alla volta e poi abbiamo verificato i risultati effettuando la media tra i due, le stesse analisi sono state fatte guardando alla nazionalità di produzione. Da questa ulteriore analisi è emerso che, come ci aspettavamo, negli Stati Uniti e nel Regno Unito vengono prodotte le serie tv generalmente più apprezzate. Al secondo posto, per gradimento troviamo le serie di origine turca grazie ad un elevatissimo livello di score medio relativo alla sentiment analysis.

# 4 Conclusioni e sviluppi futuri

Tramite questo progetto abbiamo potuto osservare come il gradimento dei fruitori delle serie televisive di Netflix sia estremamente volubile ma come sia possibile estrarre dei piccoli patterns, come per esempio l'indice di gradimento maggiore per le serie televisive statunitensi, oppure il genere preferito che nel nostro caso è risultato essere il "dramedy". All'interno dei dataset utilizzati erano presenti ulteriori caratteristiche delle serie televisive che non sono state prese in analisi dalla nostra domanda di ricerca. Si potrebbero, quindi, in futuro esplorare ulteriormente altri aspetti che rendono una serie televisiva gradita o meno al pubblico, in particolar modo agli utenti di Twitter. Abbiamo inoltre deciso di concentrare la nostra analisi unicamente sulle serie televisive, che rappresentano il contenuto maggiore all'interno della piattaforma Netflix, ma in un momento futuro essa potrebbe essere estesa anche ai film. Confidando nell'aver fatto un lavoro impegnato e nel massimo dello sforzo, gli autori si riservano di poter un giorno migliorare il workflow e l'analisi.

# Riferimenti bibliografici


[1] https://www.kaggle.com/shivamb/netflix-shows,

[2] https://www.kaggle.com/harshitshankhdhar/tv-series-dataset

[3] https://kafka-python.readthedocs.io/en/master/

[4] https://docs.tweepy.org/en/stable/

[5] https://nifi.apache.org/docs.html