DATA MANAGEMENT: NETFLIX+IMDB

Gaetano Chiriaco (882638)[1], Riccardo Porcedda (886719)[1],
Gianmarco Russo (887277)[1].

[1]Università degli studi di Milano-Bicocca, CdLM Data Science

ABSTRACT Il seguente studio si incentra sull'*acquisizione, aggregazione, integrazione, pulizia* ed *immaganizzazione* in **MongoDB** di una serie di dataset riguardanti la piattaforma di streaming **Netflix** . In particolare, sono state utilizzate diverse tecniche di *data acquisition* come il *web scraping* e le *API*. Una volta ottenuti i dati necessari per rispondere alle domande di ricerca, essi sono stati puliti ed arricchiti. Decine di attributi riguardati i titoli finiti nelle Top 10 settimanali di **Netflix** sono stati ricavati dai dati forniti da **IMDb**. Con il complesso processo di arricchimento è stato ottenuto un insieme di dati più ricco e dalla struttura più "flessibile", memorizzato nel database *document based* **MongoDB**.

Parole chiave: Data Acquisition - DBMS - Data Preparation

1 Introduzione

Netflix è una delle piattaforme di streaming più utilizzate al mondo e propone una *Top 10 settimanale* con i film e le serie TV più viste. La *Top 10* è la prima cosa che ogni utente vede nel momento in cui accede al proprio account, ed inevitabilmente influenza le scelte degli abbonati. Dal *Novembre 2021*, **Netflix** ha messo a messo a disposizione un insieme di dati contenente i dieci titoli inglesi e non-inglesi più visti ogni settimana. Le Top 10 settimanali vanno dal **04/07/2021** al **12/12/2021**, e sono divise in quattro categorie: Serie TV Inglesi, Serie TV non-inglesi e film inglesi e non-inglesi.
Nonostante sia interessante sapere quali film e serie TV sono più visti sulla piattaforma di streaming, l'insieme di dati è poco ricco di informazioni, contenendo solamente il *titolo del film*, la *posizione in classifica* ed il *numero di ore di visualizzazione*. L'obiettivo di questo studio è ottenere un dataset integrato più ricco di informazioni su ogni film, al fine di rispondere a varie domande di ricerca, come ad esempio: i film in Top 10 sono effettivamente apprezzati dagli utenti? Come varia il numero di ore di visualizzazione nelle settimane? Quali generi sono i più popolari?. Per riuscire a rispondere a queste domande si è scelto di integrare i dati di **Netflix** coi dati provenienti da **IMDb**, sito che raccoglie le recensioni degli utenti e della critica di film e serie TV, genere, data di uscita e decine di ulteriori informazioni su ogni titolo.

2 Data Acquisition

Col fine di creare un insieme di dati ricco di informazioni su ogni titolo di **Netflix** finito in Top 10, è stato necessario acquisire dati dalle due fonti, correggere le varie incongruenze, per poi unire i vari dataset necessari per rispondere alle varie domande.
Cinque dei dataset utilizzati sono stati prelevati attraverso *download* dal sito **IMDb**, che li pubblica in maniera gratuita e libera. Uno è stato invece scaricato dalla famosa piattaforma open-source **Kaggle**. Mentre il dataset riguardante i Top 10 titoli per ogni settimana è fornito direttamente da Netflix, e scaricabile su sito ***top10.netflix.com***.
Per effettuare con la maggiore accuratezza possibile le operazioni di join, con lo scraping sono stati prelevati solo gli Id dei titoli presenti ora o in passato sul catalogo Netflix. Inoltre, utilizzando l'API fornito da IMDb, sono state raccolte ulteriori caratteristiche riguardo ai film finiti in Top 10.

2.1 Dati ottenuti con Download

I dataset utilizzati ed acquisiti attraverso download sono in totale 6:

1. **Title Basics IMDB** (Clicca per download)(681,7 MB): Dataset fornito da **IMDb**, contenente tutti i film, serie TV ed altri tipi di titoli presenti nel loro database. Essendo l'unico dataset che fornisce la coppia *Id-Titolo*, è stato fondamentale in fase di integrazione per associare ad ogni film **Netflix** il proprio identificativo **IMDb**.

Tabella 1: 8.658.181 righe

tconst	titleType	primaryTitle	originalTitle	isAdult	startYear	endYear	...
tt0000001	short	Carmencita	Carmencita	0	1894	NULL	...
tt4892682	movie	A journey through time with Antony	Pei an dong ni du guo man chang sui yue	0	2015	NULL	...
tt4893588	tvEpisode	Lisa	Lisa	0	2015	NULL	...
tt9913812	tvSeries	Checkmate	Checkmate	0	2013	2015	...
tt1214796	tvEpisode	Sleeping Beauty	Sleeping Beauty	1	2002	NULL	...

- *tconst*: codice identificativo del titolo;
- *titleType*: variabile categoriale che indica se è un film o serie TV;
- *primaryTitle*: titolo internazionale del film;
- *originalTitle*: titolo originale (se diverso dal primary title);
- *isAdult*: variabile dicotomica che indica se il titolo è per adulti;
- *startYear*: anno di pubblicazione (se è una serie TV si riferisce all'uscita della prima stagione);
- *endYear*: NULL per i film, anno

dell'ultima stagione per le serie TV;

- *runtimeMinutes*: durata in minuti per i film, durata media di una puntata per le serie TV;
- *genres*: generi a cui appartiene il titolo;

2. **Ratings IMDb** (Clicca per download) (19,6 MB): Dataset utilizzato per arricchire la *Top 10* con la valutazione data dagli utenti di **IMDb**.

Tabella 2: 302.217 righe

tconst	averageRating	numVotes
tt0000001	7.9	10000
tt0000002	4.9	11222
tt0000003	6.1	34212
tt0000004	4.2	105
tt0000005	8.1	23232

- *tconst*: codice identificativo del titolo;
- *averageRating*: media voto degli utenti su IMDb;
- *numVotes*: numero di voti degli utenti su IMDb;

3. **Names IMDb** (Clicca per download) (681,7 MB): Anche questo set di dati è stato fornito da **IMDb**. Contenendo molte informazioni su ogni attore nel database, è stato fondamentale nella fase di arricchimento.

Tabella 3: 11.379.083 righe

nconst	primaryName	birthYear	deathYear	primaryProfession	knownForTitles
nm0000004	John Belushi	1949	1982	actor,soundtrack, writer	tt0078723,tt0077975, tt0072562,tt0080455
nm0000003	Brigitte Bardot	1934	NULL	actress, soundtrack, music department	tt0056404,tt0049189, tt0054452,tt0057345
nm0000019	Federico Fellini	1920	1993	writer, director, assistant_director	tt0047528,tt0056801, tt0050783,tt0071129

- *nconst*: codice identificativo dell'attore;
- *primaryName*: nome e cognome dell'attore;
- *birthYear*: data di nascita, formato(YYYY);
- *deathYear*: data di morte se esistente, altrimenti NULL;
- *primaryProfession*: le tre professioni (relative al contesto televisivo-cinematografico) più esercitate dal soggetto;
- *knownForTitles*: titoli per il quale il soggetto è famoso.

4. **Cast IMDB** (Clicca per download) (2,14 GB): Tabella associativa tra gli attori,registi e sceneggiatori ed i film. In fase di integrazione viene utilizzata per associare ogni attore ad i film in *Top 10*.

Tabella 4: 48.654.706 righe

tconst	ordering	nconst	category	job	characters
tt0000734	2	nm0000481	self	NULL	["Self"]
tt4570332	1	nm0023742	composer	NULL	NULL
tt0031902	5	nm2990971	producer	producer	NULL
tt5421110	1	nm2220901	actor	NULL	["Peter"]
tt5031001	2	nm1129491	actor	NULL	["The Boy"]

- *tconst*: codice identificativo titolo;
- *ordering*: codice id delle persone coinvolte nel titolo;
- *nconst*: codice alfanumerico identificativo della persona;
- *category*: categoria di lavoro che la persona svolge;
- *job*: titolo specifico del lavoro svolto;
- *characters*: nome del personaggio interpretato, se presente;

5. **Netflix Kaggle** (Clicca per Download) (3,4 Mb): Dataset contenente informazioni su tutti i film presenti sul catalogo **Netflix** fino al Settembre 2021. Anche in questo caso, questo insieme di dati è stato utilizzato per arricchire il dataset finale con le variabili *"date_added"* e *"release_year"*.

Tabella 5: 8.807 righe

showID	type	title	director	cast	country	...
s12	TV Show	Bangkok Breaking	Kongkiat Komesiri	[Sukollawat Kanarot,...]	NULL	...
s19	Movie	Intrusion	Adam Salky	[Freida Pinto,...]	NULL	...
s28	Movie	Grown Ups	Dennis Dugan	[Adam Sandler,..]	United States	...
s47	Movie	Safe House	Daniel Espinoza	[Denzel Washington,..]	South Africa, United States, Japan	...
s104	Movie	Shadow Parties	Yami Amodu	[Jide Kosoko,...]	NULL	...

- *showID*: codice identificativo del titolo;
- *type*: variabile categoriale che indica se è un film o serie TV;
- *title*: titolo del film o della serie TV;
- *director*: regista del film;
- *cast*: attori principali del film o della serie TV;
- *country*: paese di produzione;
- *date_added*: giorno di pubblicazione su Netflix, indicato nel formato January 01, 1970;
- *release_year*: anno pubblicazione;
- *rating*: se vietato a fasce d'età protette viene riportato il codice;

- *duration*: durata in minuti, se serie TV numero di stagioni;
- *listed_in*: categoria/e a cui appartiene nel catalogo Netflix;
- *description*: breve descrizione del titolo;

6. **Netflix Top 10** (Clicca per Download) (66 KB): Dataset fornito da **Netflix**, contenente i dieci film e le dieci serie TV più viste ogni settimana.

Tabella 6: 961 righe

week	category	weekly rank	show_title	season_title	weekly hours viewed	cumulative weeks in top10
2021-12-12	Films (English)	1	The Unforgivable	NULL	85860000	1
2021-12-05	Films (Non-English)	7	The Claus Family	NULL	2870000	5
2021-11-14	TV (English)	6	Dynasty	Dynasty: Season 4	21020000	4
2021-11-07	TV (Non-English)	1	Squid Game	Squid Game: Season 1	85000000	8
2021-10-17	Films (English)	3	Security	NULL	9470000	2

- week: settimana in cui il titolo è in Top 10;
- category: indica se serie TV o film;
- weekly_rank: posizione in classifica nella top 10;
- show_title: titolo dello show;
- season_title: nome stagione della serie tv;
- weekly_hours_viewed: totale ore di visualizzazione settimanali visualizzate;
- cumulative_weeks_in_top10: settimane cumulative nelle quali il titolo è finito in top 10.

2.2 Dati ottenuti con API

Oltre ad avere dataset pubblici e gratuiti, **IMDb** offre anche la possibilità di utilizzare delle *API* (Application Programming Interface) per ottenere informazioni aggiuntive sui titoli d'interesse. Con l'utilizzo dell'endpoint (/API/ratings/apikey/id) è stato possibile, dato un codice identificativo, di ricavare i voti di:

- *RottenTomatoes*;
- *Metacritics*;
- *TV.com*;
- *ThemovieDB*;

- *utenti IMDb.*

Vista l'elevata presenza di missing values per i voti di *TV.com* e *ThemovieDB*, si è optato di utilizzare solamente i voti di *RottenTomatoes* e *Metacritics*. Per quanto riguarda invece i voti degli utenti **IMDb**, essi risultano ridondanti in quanto sono già presenti in un dataset precedentemente scaricato. Inoltre i rating ottenibili con l'*API* sono meno aggiornati rispetto al dataset scaricabile.
L'estrazione di queste informazioni ha il solo scopo di arricchire il dataset con i voti della critica, non ottenibili in alcun altro modo. L'operazione è stata svolta utilizzando il pacchetto *requests* di *Python* per interrogare gli endpoint e *json* per manipolare il contenuto della risposta.

2.3 Dati ottenuti con web scraping

Dato che la ricerca è incentrata solo sui titoli nel catalogo **Netflix**, si è dapprima utilizzato un altro endpoint delle *API* offerta da **IMDb** (Company/id) per trovare solo i titoli effettivamente passati o attualmente presenti su Netflix. Questa soluzione presenta però un limite: restituisce solamente i primi 50 titoli. Vista la necessità di ottenere l'intero catalogo **Netflix** sono state utilizzate tecniche di web scraping per "aggirare" i limiti dell'*API*. La libreria di *Python Beautiful Soup*[1] è stata utilizzata per lo scraping, mentre la libreria *selenium*[2] per lo scorrimento tra le pagine della ricerca web. Con lo scraping sono stati estratti il *titolo* e l'*identificativo* **IMDb** di tutti i prodotti presenti nella sezione **Netflix** di **IMDb**. Lo scraping è stato eseguito sulla pagina **IMDb** in lingua italiana.

Nella figura sottostante è mostrato un esempio della pagina su cui è stato applicato lo scraping ed i due attributi di interesse.

3 Data Enrichment & Cleaning

I sei dataset ottenuti con il *download* ed i due ottenenti attraverso *API* e *web scraping* sono difficilmente utilizzabili se non corretti, armonizzati e integrati corret-

tamente. Con l'utilizzo di tecniche per l'integrazione, matching totale e parziale e risoluzione dei conflitti, le otto diverse fonti di dati sono state condensate in un unico modello più complesso descritto approfonditamente nella sezione 4.

Di seguito sono riportati i passaggi (in forma testuale e grafica) volti a "raffinare" e integrare tutti i dati precedentemente descritti:

1. Il primo passo è stato l'unione (sulla base dell'*identificativo IMDb*) tra **"Title Basics"** ed i dati ottenuti attraverso web scraping (contenenti le coppie *Id-Titolo*). I titoli dei dati ottenuti attraverso *web scraping* sono un sottoinsieme dei titoli presenti in **Title Basics**. I dataset sono stati uniti per effettuare una prima scrematura dei titoli omonimi, in quanto il risultato contiene solamente titoli sul catalogo **Netflix**, ma arricchiti con le informazioni di **IMDb** . Ulteriori campi sono stati aggiunti effettuando una *Left Outer Join* con il dataset **"Netflix Kaggle"**. Questa operazione è stata svolta due volte, sia con *primaryTitle* che con *originalTitle*, in modo da scongiurare che un film non venga accoppiato correttamente solo a causa della diversa lingua in cui il titolo è scritto (caso di sinonimia).

2. Prima di procedere con ulteriori integrazioni e arricchimenti è necessario gestire eventuali incongruenze, omonimie e problemi relativi alla *data quality*. In particolare:

   - tra i record duplicati vengono eliminati i record con start_year=NULL, in quanto si tratta di titoli non ancora distribuiti e di conseguenza di certo non finiti in Top 10;
   - sono stati selezionati tutti i titoli duplicati, ovvero tutti i film o le serie TV che hanno degli omonimi. È fondamentale identificare quale tra i diversi omonimi sia il titolo che davvero interessa, per evitare che nelle integrazioni successive vengano associate informazioni sbagliate ad uno o più titoli;
   - tra i record duplicati engono rimossi i titoli ai quali non è stato assegnato il tipo corretto (film matchati con serie tv o viceversa);
   - tra i record duplicati vengono rimossi i titoli per cui *startYear* e *releaseYear* differiscono di più di un anno;
   - se sono ancora presenti casi di omonimia, viene semplicemente scelto il titolo più recente.

   Il dataset ottenuto è riportato nel grafico in figura alla pagina successiva con il nome **"Netflix Complete"**.

3. Nella terza fase vengono effettuate nuovamente due join (una su *originalTitle* e l'altra su *primaryTitle*) tra **"Netflix Complete"** ed il dataset **"Netflix Top 10"**. Anche in questo caso vengono utilizzate due variabili per effettuare le join

col fine di minimizzare gli errori di match. Il risultato è **"Top 10 Enriched"**, composto dalla posizione in classifica in una determinata settimana, nome del film, codice identificativo **IMDb** e un insieme di altre informazioni ricavate dai dataset utilizzati per l'arricchimento. Come risultato di questo passaggio si hanno 217 film senza Id **IMDb** e di conseguenza senza i dati di arricchimento.

4. Per risolvere i problemi di *missing match* è stato riutilizzato il dataset **"Title Basics"**. Il dataset **"Top 10 Netflix"** non è stato direttamente unito a **"Title Basics"** per scongiurare il matching tra film omonimi. Tra i 217 film senza Id, per 84 di essi è stato trovato un match univoco nel dataset **"Title Basics"**. In questi casi si è assunto che non si trattasse di un caso di omonimia. Per i film per cui si è avuto più di un match, i conflitti sono stati rimossi seguendo le cinque operazioni del passo 2.

5. Dopo l'ultimo passaggio 133 titoli non avevano ancora un match. Per risolvere il problema è stata effettuata un'ulteriore join tra **"Top 10 Enriched"** e i dataset **"Netflix IMDb Scraping** e **"Title Basics"** utilizzando la *Fuzzy Search*[1]. Essa è stata utilizzata due volte sul dataset **"Title basics"** (soglia di 0.9 su *originalTitle* e di 0.8 su *primaryTitle*) e due volte sui dati ottenuti con lo scraping (0.7 in entrambi i casi, una su *primaryTitle* e una su *originalTitle*).

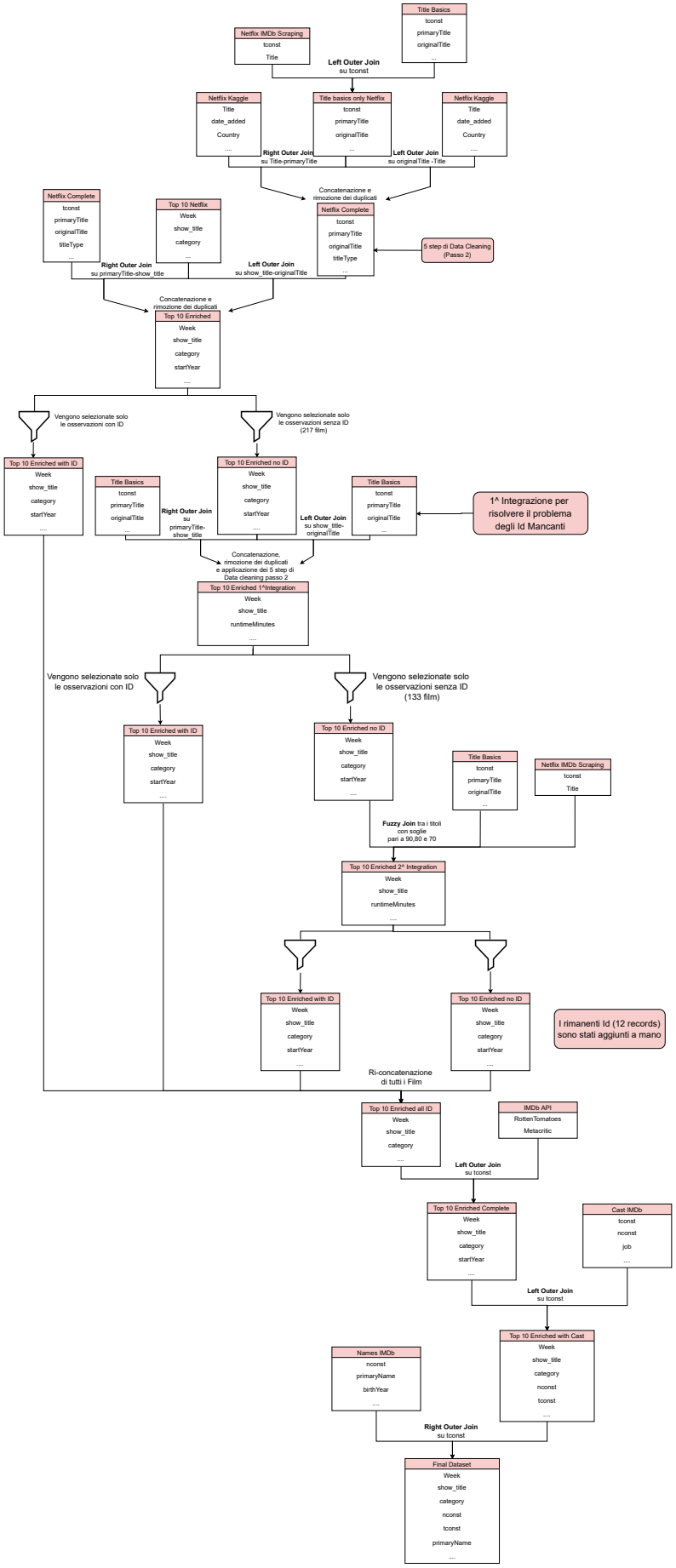
6. Dopo aver applicato anche un matching parziale, ancora per 12 titoli non è stata trovata una corrispondenza. Questi ultimi sono stati quindi corretti manualmente.

7. Una volta ottenuto il dataset finale, privo di omonimi e titoli senza Id, è stato integrato con i voti della critica ottenuti tramite l'*API* di **IMDb**.

8. Dopodiché è stata riscontrata e risolta un'ulteriore criticità: due titoli **Netflix** presentano due pagine **IMDb** ciascuno (con 2 diversi Id). Probabilmente una delle due si riferisce ad una pagina provvisoria non più aggiornata. Tra le due pagine riferite allo stesso film è stata considerata quella con il numero di voti degli utenti più elevato.

9. Infine, il dataset risultante è stato nuovamente integrato con le informazioni su attori e registi ricavate dal dataset **"Names IMDb"**. Per ogni titolo appartenente alla Top 10 di **Netflix**, è stata aggiunta una lista degli attori principali e del regista, arricchita con alcune principali informazioni su di essi.

[1] Fuzzy Search [3] è un algoritmo che permette di confrontare delle stringhe per determinarne la similarità. La caratteristica fuzzy [4] permette di impostare delle soglie di 'certezza' (variabile tra 0 incertezza totale, a 100 certezza assoluta). Viene perciò usata per accoppiare i titoli a tutti gli effetti identici, ma che per qualche motivo hanno delle piccole differenze sintattiche nel titolo.


Title Basics
tconst
primaryTitle
originalTitle
...
Netflix IMDb Scraping
tconst
Title
Left Outer Join
su tconst
Netflix Kaggle
Title
date_added
Country
....
Title basics only Netflix
tconst
primaryTitle
originalTitle
...
Netflix Kaggle
Title
date_added
Country
....
Right Outer Join
su Title-primaryTitle
Left Outer Join
su originalTitle -Title
Concatenazione e
rimozione dei duplicati
Netflix Complete
tconst
primaryTitle
originalTitle
titleType
...
Top 10 Netflix
Week
show_title
category
...
Netflix Complete
tconst
primaryTitle
originalTitle
titleType
...
5 step di Data Cleaning
(Passo 2)
Right Outer Join
su primaryTitle-show_title
Left Outer Join
su show_title-originalTitle
Concatenazione e
rimozione dei duplicati
Top 10 Enriched
Week
show_title
category
startYear
....
Vengono selezionate solo
le osservazioni con ID
Vengono selezionate solo
le osservazioni senza ID
(217 film)
Top 10 Enriched with ID
Week
show_title
category
startYear
....
Title Basics
tconst
primaryTitle
originalTitle
...
Top 10 Enriched no ID
Week
show_title
category
startYear
....
Title Basics
tconst
primaryTitle
originalTitle
...
1^ Integrazione per
risolvere il problema
degli Id Mancanti
Right Outer Join
su
primaryTitle-
show_title
Left Outer Join
su show_title-
originalTitle
Concatenazione,
rimozione dei duplicati
e applicazione dei 5 step di
Data cleaning passo 2
Top 10 Enriched 1^Integration
Week
show_title
runtimeMinutes
....
Vengono selezionate solo
le osservazioni con ID
Vengono selezionate solo
le osservazioni senza ID
(133 film)
Top 10 Enriched with ID
Week
show_title
category
startYear
....
Top 10 Enriched no ID
Week
show_title
category
startYear
....
Title Basics
tconst
primaryTitle
originalTitle
...
Netflix IMDb Scraping
tconst
Title
Fuzzy Join tra i titoli
con soglie
pari a 90,80 e 70
Top 10 Enriched 2^ Integration
Week
show_title
runtimeMinutes
....
Top 10 Enriched with ID
Week
show_title
category
startYear
....
Top 10 Enriched no ID
Week
show_title
category
startYear
....
I rimanenti Id (12 records)
sono stati aggiunti a mano
Ri-concatenazione
di tutti i Film
Top 10 Enriched all ID
Week
show_title
category
....
IMDb API
RottenTomatoes
Metacritic
Left Outer Join
su tconst
Top 10 Enriched Complete
Week
show_title
category
startYear
....
Cast IMDb
tconst
nconst
job
....
Left Outer Join
su tconst
Top 10 Enriched with Cast
Week
show_title
category
nconst
tconst
....
Names IMDb
nconst
primaryName
birthYear
....
Right Outer Join
su tconst
Final Dataset
Week
show_title
category
nconst
tconst
primaryName
....

4 Data Modelling: MongoDB

Dopo aver ottenuto ed integrato tutti i dati necessitati è stato necessario scegliere il modello di storage più adatto alla struttura delle informazioni a disposizione. Data la presenza di ciclicità nei dati, numero di attributi variabile ed attributi multi-valore, l'approccio documentale è risultato il più adatto al problema studiato. L'implementazione dello schema di storage documentale è stata effettuata utilizzando **MongoDB**.

MongoDB[5] è un *DBMS NoSQL* di tipo documentale. Soddisfa le proprietà **C** (consistenza) e **P** (tolleranza al partizionamento) del **CAP theorem**[6]. È basato su formato dati *BSON* (Binary JSON), adatto all'interscambio dei dati sul web. Essendo un modello *schema-less*, non impone vincoli sulla struttura dei dati. È composto da documenti, contenuti in collezioni a loro volta contenute in un database. Prevede un'indicizzazione dei documenti durante in loro inserimento per agevolarne il fetching nelle successive ricerche.

I dati descritti nella Sezione 2 ed arricchiti nella Sezione 3 sono stati formattati in una struttura adatta ad un modello documentale come segue:

```
{
  "id": "tt0000001",
  "Title": "Film name",
  "Type": "Film(English)",
  "Runtime": 110,
  "Genres":["Action", "Comedy"],
  "StartYear": 2019,
  "IsAdult": 0,
  "DateAdded":"2021-11-29",
  "Description":"...",
  "Country":["United States", "Australia"],
  "Rating": {
    "User": {
        "Vote":6.5,
        "NumVotes": 12500},
    "Critics":{
        "Metascore": 5.5,
        "RottenTomatoes": 6.0}
    },
   "Top10":{
   "Week":["2021-12-12", "2021-12-05"],
   "Place":[1,3],
   "WeeklyHours":[1230000, 900000]},
   "Crew":{
       "Directors":[{
```

```
        "Name": "Director's name",
        "BirthYear": 1959,
        "PrimaryProfessions":["director","actor"],
        },
        {
        "Name": "Director's name",
        "BirthYear": 1920,
        "PrimaryProfessions":["director","composer"],
        }],
       "Actors":[{
        "Name": "Actor's name",
        "BirthYear": 1971,
        "PrimaryProfession": ["actor"],
        "KnownForTitles":["tt0000001","tt0000002"]
        }]
        },
    "EndYear": NULL
}
```

I dati relativi ai film e alle serie TV finiti nelle *Top 10* di **Netflix** sono stati formattati in documenti e caricati su un **server AWS** tramite la piattaforma cloud di **MongoDB**, che permette l'interazione dalla rete, non solo in locale. Una struttura come quella rappresentata comporta una considerevole diminuzione dello spazio di memoria necessario: infatti, se uno dei film già presenti nella collezione continua ad occupare la *Top 10* in futuro, basta solamente aggiornarlo senza alcun necessità di inserire un nuovo documento o una nuova riga.

5 Valutazione di Qualità

La valutazione di qualità è uno step necessario per identificare le qualità ed i difetti dei vari dataset utilizzati. La qualità di un insieme di dati può essere valutata sulla base di quattro caratteristiche principali:

- *Currency*
- *Consistency*
- *Completeness*
- *Accuracy*

Tre di queste caratteristiche (*Currency*, *Consistency* e *Completeness*) sono state studiate sul dataset contenente i singoli film, il dataset contenente gli attori ed i registi ed infine il dataset contenente le varie *Top 10* settimanali di **Netflix**.

5.1 Currency

Il dataset rappresenta solo una sezione recente di tutte le *Top 10* stilate da **Netflix** nel corso degli anni. Come detto nell'introduzione, solo dal *Novembre 2021* la piattaforma ha reso pubblica e consultabile questa risorsa. La *Top 10* più "vecchia" a disposizione risale al *4 Luglio 2021*, mentre la più recente è del *17 Dicembre 2021*. Ogni giovedì **Netflix** aggiunge dati su un'ulteriore settimana.

I dati attualmente a disposizione ed inclusi nel dataset finale sono aggiornati al *17 Dicembre 2021*, ma la *currency* del dataset diminuisce di settimana in settimana a causa dell'uscita di nuovi film e serie TV e di nuove *Top 10*.

5.2 Consistency

Uno degli ostacoli principali incontrati nella fase di arricchimento è stata la scarsa coerenza tra i vari dataset utilizzati. Nel dataset finale è stata garantita la coerenza tra tre principali tipi di dati, che inizialmente differivano tra le varie fonti di dati per *formattazione*, *scala* e *sintassi*:

- **Consistency tra le date**: Nel dataset scaricato da **Kaggle** le date erano espresse nel formato *"January 01, 1970"*, mentre nei dataset estratti da IMDb nel formato *"1970-01-01"*. Durante l'arricchimento, tutte le data sono state trasformate nel formato *"1970-01-01"* per dare uniformità.
- **Consistency tra i titoli**: L'inconsistenza tra i titoli è stato uno dei più grandi problemi da risolvere per garantire una corretta unione delle varie fonti dei dati. Tra i vari dataset lo stesso titolo poteva essere scritto in tre modi differenti:
  - Nel dataset **"Netflix Top 10"** e **"Title Basics"**, il titolo è scritto in lingua inglese. Ad esempio la serie TV *"La casa di Carta"* viene nominata *"Money Heist"*;
  - Nei dati ottenuti con lo scraping, i titoli sono in lingua italiana. In questo dataset, *"La casa di Carta"* viene nominata in questo modo;
  - Nel dataset **Kaggle**, alcuni dei titoli sono scritti in inglese, mentre altri in lingua originale. In questo caso, la serie TV *"La casa di Carta"* può essere intitolata sia *"Money Heist"* che *"La casa de Papel"*.

  Inoltre, lo stesso titolo può essere scritto con alcune lettere maiuscole o con simboli di punteggiatura diversa. Dove possibile, i titoli sono stati tradotti in inglese, utilizzando gli accoppiamenti *primaryTitle-originalTitle*. Inoltre, tutti i titoli sono stati riscritti in minuscolo.
- **Consistency tra i ratings**: I rating di **IMDb**, **Metascore** e **RottenTomatoes** sono tutti e tre rappresentati in scale diverse. Il *rating* **IMDb** assume valori

che vanno da 0 a 10, **Metascore** da 0 a 100, mentre **RottenTomatoes** descrive il rating come un valore da "0%" a "100%". Ad ogni modo, si è preferito mantenere questa inconsistenza per conservare la fedeltà al dato originale (in caso si volesse effettuare dei controlli con i siti di Metascore e RottenTomatoes).

5.3 Completeness

Sui quattro dataset è stata calcolata l'**attribute completeness**, riportata nella Tabella 7 e la **table completeness**,riportata nella Tabella 8.

Tabella 7: Completezza per Attributo

Feature	Percentuale di NULL	Numero di NULL	Numero di osservazioni
Id	0%	0	350
Title	0%	0	350
Type	0%	0	350
Runtime	8%	28	350
Genres	0%	0	350
isAdult	0%	0	350
StartYear	0.2%	1	350
NumVotes	0%	0	350
Vote	0.8%	3	350
DateAdded	43%	150	350
Description	43%	150	350
Country	62%	219	350
Metascore	59%	205	350
RottenTomatoes	60%	212	350
Name (Actor)	0%	0	1815
BirthYear (Actor)	38%	694	1815
PrimaryProfession (Actor)	6%	102	1815
KnownForTitles (Actor)	0.7%	13	1815
Name (Director)	0%	0	225
BirthYear (Director)	52%	116	225
PrimaryProfession (Director)	0%	0	225
KnownForTitles (Director)	0%	0	225
Week (Top 10)	0%	0	960
WeeklyHours (Top 10)	0%	0	960
Place (Top 10)	0%	0	960

- Per **Vote** tre valori sono NULL perchè nessun utente di **IMDb** ha dato una valutazione per questi titoli.
- I valori nulli per **DateAdded** e **Description** sono causati dall'assenza di 150 finiti in *Top 10* nel dataset **"Netflix Kaggle"**.
- Per gli attributi **Metascore** e **RottenTomatoes** si hanno rispettivamente 205 e 212 valori nulli, siccome questi due siti non hanno dato una valutazione a tutti questi titoli.

Tabella 8: Completezza tabellare

Dataset	Percentuale di NULL	Numero totale di NULL	Numero totale di valori
Film	20%	968	4900
Attori	11%	809	7260
Registi	13%	116	900
Top 10	0%	0	2880

6 Analisi Esplorative

Il dataset definitivo, ottenuto attraverso arricchimenti e correzioni per ottenere una maggiore qualità, è composto da **350 documenti**. Essi rappresentano i 350 film e serie TV distinti finiti in *Top 10* da Luglio a Dicembre. Il numero di attributi di cui un documento è composto varia a seconda delle caratteristiche del titolo:

- Per i **film** il numero di attributi è pari a:

  $$17 + 5(A + R)$$

  dove 17 rappresenta il numero di attributi fissi, A rappresenta il numero di attori del film e R rappresenta il numero di registi.

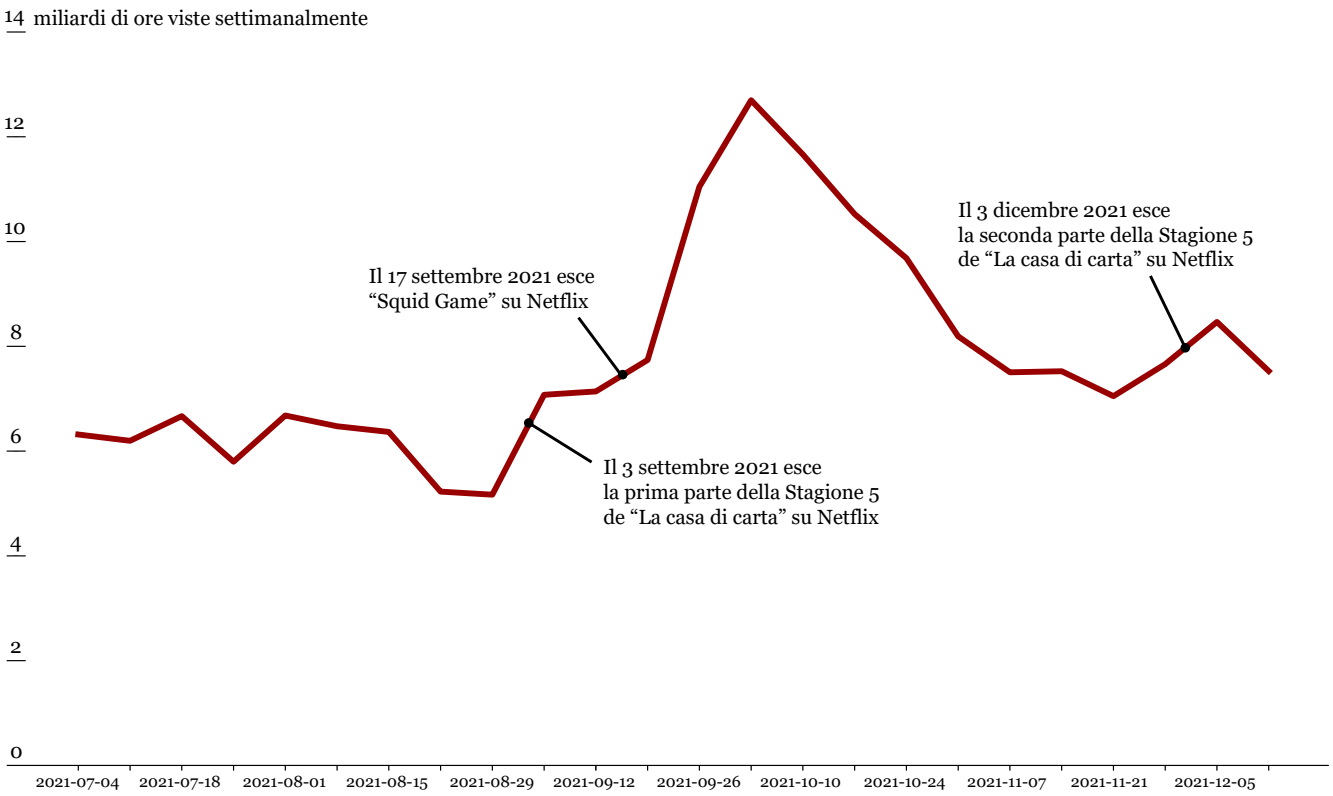
- Per le **serie TV** il numero di attributi è pari a:

  $$17 + 5(A + R) + 1$$

  L'unico attributo in più rispetto ai film è *EndYear*, che non ha senso di esistere nei film, in quanto è un valore che indica l'anno di chiusura di una serie TV.

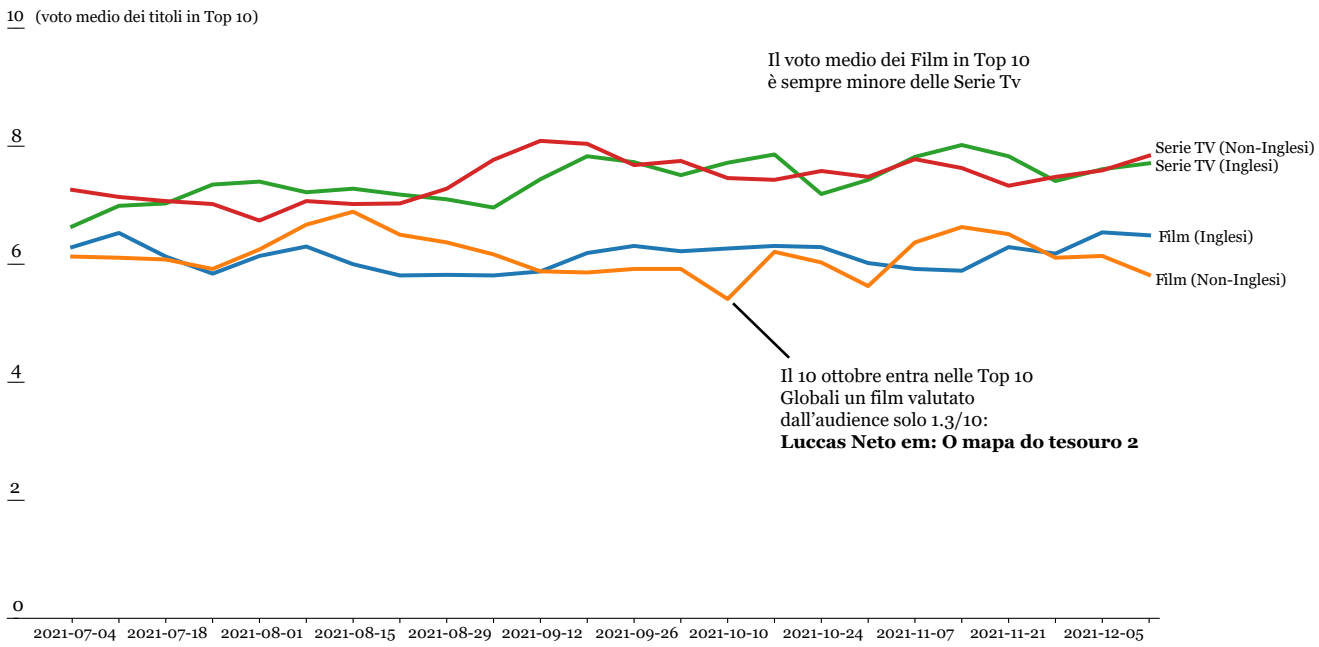
Con l'integrazione tra i dati di **Netflix** e di **IMDb** è possibile esplorare ed analizzare le relazioni tra *numero di ore viste*, *posto in classifica*, *valutazione degli utenti* e *dei critici* e decine di altre variabili.

Nella figura sottostante viene riportato l'andamento nelle settimane del *numero di ore di visualizzazione totali* su **Netflix**.



- Il *numero di ore di visualizzazione settimanali* non scende mai sotto i **quattro miliardi**.
- Anche l’uscita di un singolo titolo può portare ad un fortissimo impatto sulla quantità di ore spese sulla piattaforma, come nel caso di **"Squid Game"**

Nel grafico riportato in figura sottostante, è stato analizzato l’andamento del voto medio degli utenti nel tempo. Dopo aver ricavato il voto medio di ogni titolo, è stato calcolato il voto medio dei titoli presenti nella *Top 10* in una determinata settimana.



- Le serie TV (Inglesi e Non-Inglesi) sono sempre più apprezzate dall'audience rispetto ai film;
- il voto medio rimane pressoché stabile nelle varie settimane.

7 Conclusioni

Il lavoro ha previsto il download di 6 dataset diversi, uniti ad un integrazione tramite API e una tramite web scraping. I dati sono stati poi accorpati e puliti per eliminare informazioni non utili alla ricerca, ovvero analizzare i dati delle Top 10 Netflix. è stato modellato uno schema di dati utilizzando mongoDB ed è stato caricato il risultato del dataset finale sul dbms. Una volta ottenuto il dataset finale sono state inoltre compiute delle analisi esplorative, al fine di analizzare ciò che è stato prodotto. Il risultato finale è stato avere un database pronto all'uso e in grado di fornire i dati utile alla risoluzione delle domande di ricerca.
Si evidenzia infine che il workflow di questa ricerca è stato realizzato avendo in mente che dataset scaricabili ed API saranno costantemente aggiornati, permettendo dunque di aggiungere facilmente nuovi dati e arricchirli, dando vita ad un dataset che acquista sempre più valore nel tempo e che potrà rispondere ad un numero crescente di domande di ricerca.

Sitografia

[1] *Beautiful Soup. Libreria per web scraping*. URL: https://www.crummy.com/software/BeautifulSoup/bs4/doc/.

[2] *Selenium Webdriver. Interazione con pagine web*. URL: https://selenium-python.readthedocs.io/.

[3] *Fuzzy Search. Libreria python*. URL: https://pypi.org/project/fuzzywuzzy/.

[4] *Fuzzy Search. Spiegazione Teorica*. URL: https://en.wikipedia.org/wiki/Approximate_string_matching.

[5] *MongoDB. NoSQL DBMS*. URL: https://www.mongodb.com/.

[6] *Teorema CAP. DBMS*. URL: https://web.archive.org/web/20111122142126/http://lpd.epfl.ch/sgilbert/pubs/BrewersConjecture-SigAct.pdf.