Netflix and chill: studio degli show che arrivano alla top ten di Netflix. Relazione finale del progetto per il corso di Data Management and Visualization

GIOVANNI DE FEUDIS[1], CAMILLA DI MARTINO[2], AND IMAN RAS[3]

[1] 820602 CdLM Data Science, Università degli Studi di Milano-Bicocca
[2] 873147 CdLM Data Science, Università degli Studi di Milano-Bicocca
[3] 812509 CdLM Data Science, Università degli Studi di Milano-Bicocca

Compiled June 16, 2021

La grande diffusione delle piattaforme di streaming ha generato nel mondo dell'intrattenimento un profondo cambiamento negli ultimi anni. La crescita di questi servizi è dovuta principalmente alla loro facilità e comodità di fruizione. Tra le varie piattaforme disponibili abbiamo deciso di analizzare Netflix, che è la più in voga al momento ed è anche la più amata dagli utenti.
Periodicamente su Netflix vengono aggiornati i contenuti, mettendo così a disposizione degli utenti una scelta sempre più ampia. Inotre, Netflix ha creato la Top Ten dei titoli più popolari nei singoli paesi ogni giorno. Noi ci siamo chiesti cosa guardino gli italiani su Netflix e quali caratteristiche abbiano i film e le serie-tv che sono finite in Top Ten negli ultimi mesi, in particolare nel periodo tra il 1 Dicembre 2020 e il 31 Marzo 2021.
Tale progetto si pone dunque l'obiettivo di misurare la popolarità dei film e delle serie-tv analizzando la loro permanenza e la loro posizione nelle due Top Ten: La Top Ten generale e la Top Ten delle due diverse categorie (serie-tv e film). Inoltre confrontiamo la popolarità degli show con altre caratteristiche come il genere, la trama, il cast o la qualità generale dello spettacolo.

CONTENTS

1. INTRODUZIONE

La nostra ricerca verte proprio sui titoli maggiormente di successo presenti su Netflix, di cui vogliamo studiare la popolarità tra il pubblico e la qualità come prodotto artistico.
La popolarità è misurata nella presenza, in termini di numero di giorni, nelle classifiche sopra citate che riportano gli show più visti; essa viene calcolata attraverso un opportuno indice

che introdurremo in seguito. La qualità artistica invece abbiamo deciso di misurarla tramite i voti (*rating*) che ciascun titolo ha ottenuto sul sito Internet Movie Database (IMDb).
Il sito raccoglie i voti lasciati dagli utenti, che pur non costituendo recensioni professionali sono un indicatore della qualità percepita dal grande pubblico. Infatti IMDb è molto popolare quindi ci si aspetta che molte persone commentino, specialmente se parliamo di titoli famosi, come quelli presenti nelle classifiche di Netflix che abbiamo raccolto.
Il focus della nostra ricerca è stato capire quali caratteristiche debba avere un film, o una serie tv, per finire in classifica su Netflix; scopo che abbiamo perseguito studiando a fondo i titoli che sono stati in classifica dal 1 Dicembre 2020 al 31 Marzo 2021. La nostra ricerca ha indagato esclusivamente le classifiche di Netflix Italia. Abbiamo dunque raccolto i titoli dagli show presenti nelle varie Top Ten nel periodo sopra citato, e ne abbiamo studiato, oltre alle già citate popolarità e qualità secondo IMDb, tutte le caratteristiche, come il paese di produzione, la durata, i premi vinti, il genere e l'accoglienza da parte della critica.

La raccolta dati è stata resa possibile grazie all'utilizzo dell' API del sito FlixPatrol che, riporta ogni giorno tutte le classifiche di Netflix; inoltre la raccolta delle caratteristiche dei singoli show è stata fatta facendo scraping su Wikipedia; mentre per quanto riguarda i dati di IMDb ci sono stati resi disponibili sotto forma di dataset tabellari dai gestori del sito.
Da ogni fonte è stato ottenuto un dataset, ognuno dei quali è stato inserito all'interno di un database MongoDB tramite Mongo Atlas. Terminata la fase di raccolta dati, si è proseguito con la fase di integrazione dei dataset sfruttando il linguaggio Python. Il dataset integrato è stato oggetto di una fase di pulizia e *preprocessing*. I dati processati e puliti sono stati utilizzati per le analisi finali e le visualizzazioni sono state realizzate mediante il software Tableau.

Delle tre V dei Big Data il nostro progetto ha affrontato la:

- **Velocità**: le tre top ten di Netflix si aggiornano quotidianamente, perciò si qualificano come dati "veloci", precisamente con un periodo di aggiornamento di un giorno. Queste tre classifiche sono: i dieci film più visti del giorno (classifica dei film), le dieci serie tv più viste del giorno (classifica serie tv) e infine la classifica generale, ossia i dieci show più visti considerando sia serie tv sia film. Utilizzando l'API del sito FlixPatrol, abbiamo raccolto le tre classifiche Netflix di ogni giorno compreso tra il 1 Dicembre 2020 e il 31 Marzo 2021, costruendo un sistema per la raccolta di informazioni disaccoppiato dalla memorizzazione delle stesse. I dati sono stati salvati in formato JSON, e successivamente convertiti in CVS per la fase di memorizzazione. Per i dettagli rimandiamo ai paragrafi successivi.

- **Varietà**: i dati "veloci" costituiti dalle classifiche di ogni giorno su Netflix sono stati integrati con due fonti di dati: il database IMDb, che, come già accennato, per ogni show riporta la valutazione media data dagli utenti, e che consideriamo come indice di **qualità** del titolo in questione; e le informazioni prese da Wikipedia tramite *scraping*, che descrivono in modo esaustivo ogni film o serie tv del database. Tra i dati presi da quest'ultima fonte abbiamo il genere, il paese di produzione, la durata o il numero di episodi e molto altro per ogni serie tv o film considerato.

2. CALCOLO DELL'INDICE DI POPOLARITÀ

Date le classifiche di ogni giorno per quattro mesi, bisogna trovare un modo per identificare quali film o serie tv siano stati i più visti tra tutti, non solo per un giorno, ma globalmente nei quattro mesi a cui si riferiscono i dati.

Perciò abbiamo pensato al calcolo di un indice, che abbiamo chiamato **indice di popolarità** che calcoli la popolarità generale di un titolo in un periodo esteso.
Per ogni show (serie tv oppure film) f consideriamo preliminarmente la sommatoria:

$$\sum_{g \in G} (10 - rank_{C,g}(f))$$

dove G è l'insieme dei giorni considerati (in tutto 121 giorni, dal 1 Dicembre al 31 Marzo), e $rank_{C,g}(f)$ è il *ranking* di f nel giorno g, ossia la posizione di f all'interno della classifica C nel giorno g. In particolare considerando la classifica C (che può essere quella generale, quella dei film o quella delle serie tv) al giorno g:

$$rank_{C,g}(f) = \begin{cases} 0 & \text{se } f \quad \text{è primo in classifica} \\ 1 & \text{se } f \quad \text{è il secondo} \\ \dots & \\ 9 & \text{se } f \quad \text{è l'ultimo, cioè il decimo} \\ 10 & \text{se } f \text{ non compare in classifica} \end{cases}$$

Per esempio: se il titolo "Lupin" era in prima posizione all'interno della classifica delle sole serie tv il giorno 1 Gennaio allora:

$$rank_{(serie\,tv,\,1/1)}(\text{Lupin}) = 0$$

In questo modo per i giorni in cui il titolo f è in classifica più la posizione è alta più la sommatoria sopra riportata aumenta, mentre i giorni in cui f non è in classifica non danno nessun contributo, dato che il termine della sommatoria relativo a quel giorno sarà: $10 - 10 = 0$.
A questo punto siamo pronti per presentare l'**indice di popolarità relativo alla classifica C**:

$$I_C(f) = \frac{\sum_{g \in G}(10 - rank_{C,g}(f))}{10 * |G|}$$

dove $|G|$ indica la cardinalità dell'insieme G, ossia il numero di giorni considerati.
Per esempio: consideriamo ancora la serie tv "Lupin", e consideriamo la top ten delle sole serie tv.
Ipotizziamo che Lupin sia stata prima in classifica il 1 gennaio e il 2 gennaio, seconda il 3 gennaio, quarta il 4 e infine che il 5 gennaio non fosse più presente in classifica. Considerando il periodo di tempo dal 1 al 5 gennaio allora abbiamo:

$$I_{tv}(\text{Lupin}) = \frac{(10-0)+(10-0)+(10-1)+(10-3)+(10-10)}{10*5}$$

$$= \frac{10+10+9+7+0}{10*5} = 0,72$$

In questo caso abbiamo considerato un periodo di 5 giorni, nel progetto, come già accennato, i giorni totali sono 121.
L'indice dipende ovviamente dalla classifica: uno stesso titolo ha indici diversi se consideriamo una classifica o l'altra, inoltre i film non possono ovviamente comparire nella classifica delle serie tv e viceversa.
Mettendoci nell'esempio di prima, in cui Lupin ha $I_{tv}(\text{Lupin}) =$

0,72 per il periodo dal 1 al 5 Gennaio, supponiamo che lo show fosse anche primo in classifica generale il 1 gennaio, secondo il 2, quinto il 3, decimo il 4 e poi non fosse più in classifica.
Allora si avrebbe:

$$I_{gen}(\text{Lupin}) = \frac{10+9+6+1+0}{10*5} = 0,52$$

Quindi Lupin ha indici diversi a seconda della classifica. Non ha senso considerare il suo indice per la classifica dei film perché, ovviamente, essendo una serie tv non vi compare mai.

Segue dalla definizione che **l'indice relativo ad una classifica è compreso tra 0 e 1**, dove 0 è il valore minimo (corrispondente a un film che non è mai stato nella classifica C), mentre 1 è il valore massimo: se un film o una serie tv f ha $I_C(f) = 1$ significa che per ogni giorno del periodo considerato è stato primo nella classifica C.
Siamo pronti per definire l'**indice di popolarità** dello show f:

$$I(f) = I_{gen}(f) + I_{tipo}(f)$$

dove il periodo considerato sono i quattro mesi di raccolta di dati, e l'indice $I_{tipo}(f)$, che nel database finale chiamiamo anche **indice tipo**, è l'indice relativo alla classifica delle serie tv se f è una serie tv, altrimenti è l'indice relativo alla classifica dei film. In altre parole:

$$I_{tipo}(f) = \begin{cases} I_{\text{film}}(f) & \text{se f è un film} \\ I_{\text{serie tv}}(f) & \text{se f è una serie tv} \end{cases}$$

L'indice di popolarità è compreso **tra 0 e 2**.

Nel dataset finale abbiamo salvato separatamente per ogni titolo $I_{gen}(f)$, chiamato **indice generale** perché relativo alla classifica generale, e $I_{tipo}(f)$ nella colonna appunto chiamata "indice tipo".

Osserviamo infine che, tra tutti i programmi che abbiamo, solo pochi arrivano in classifica generale; questo implica che molti titoli del database sono stati raccolti perché sono comparsi nella classifica relativa al proprio tipo, e mai in quella generale (è il caso di moltissimi film minori). In questo caso, se il film (o serie tv) f non compare in nessun giorno nella classifica generale allora:

$$I_{gen}(f) = 0$$

È chiaro che tutti gli show nel nostro database sono stati in almeno una delle tre classifiche (altrimenti non sarebbero nel database) perciò nessuno show può avere indice di popolarità esattamente pari a zero, al massimo può capitare che nella somma $I_{gen}(f) + I_{tipo}(f)$ uno dei due termini sia uguali a zero (e se ciò accade, molto probabilmente è l'indice generale a essere uguale a zero, dopotutto è più difficile entrare nella classifica generale piuttosto che nella classifica specifica).

3. VELOCITÀ

A. Acquisizione dei dati

L'acquisizione dei dati utili è stata ottenuta attraverso tre diverse fasi:

1. in primo luogo,in riferimento alle **Top Ten di Netflix** di film e serie-tv sono stati scaricati i vari dati tramite API (application programming interface), fornita dal sito FlixPatrol. Successivamente, tramite una semplice configurazione, è stato possibile ottenere i dati di interesse. Dall' 1 Dicembre 2020 al 31 Marzo 2021, è stata indirizzata una chiamata all' **endpoint** ogni 24 ore, tramite un'apposita request.
Il risultato di tali operazioni ha condotto alla produzione di un documento in formato JSON, il cui contenuto consiste in:
   (a) Una serie di informazioni utili riferite all'identità dei dati esaminati;
   (b) Le date di formazione delle Top Ten;
   (c) Tre liste contenenti varie informazioni inerenti a Film e serie-tv presenti nelle Top Ten.
2. Questo flusso di dati è stato poi incanalato attraverso **Apache Kafka**, una piattaforma di stream processing. Tutte le procedure sono state effettuate su una macchina in locale con sistema operativo Ubuntu, distribuzione GNU/Linux. Dopo l'istallazione in locale di tutto l'environment Kafka, si è passati alla configurare di **Zookeeper**:

   ```
   bin/zookeeper-server-start.sh config/zookeeper.properties
   ```

   Successivamente il **Server Kafka** è stato inizializzato:

   ```
   bin/kafka-server-start.sh config/server.properties
   ```

   Quindi è stato creato il **topic**:

   ```
   bin/kafka-topics.sh --create --topic exam --bootstrap-server localhost:9092
   ```

3. Alla fine tramite l'utilizzo della libreria **Kafka-Python**, il file JSON già serializzato è stato inviato tramite il **Producer** alla coda Kafka (vedi figura 2).

   I dati caricati dal Producer all'interno della coda Kafka sono stati poi prelevati dal **Consumer**.

Fig. 1. La figura mostra il procedimento di configurazione del **Producer** mediante l'uso della libreria Kafka-Python

```
In [ ]: from kafka import KafkaProducer
        import json
        import time
        import requests
        from pprint import pprint
        from pymongo import MongoClient
        from datetime import date
```

la funzione date.today() mi ridà la data di oggi, che poi verrà aggiunta nel ciclo while che ha un timer di 24 ore, quindi ogni 24 ore la data si aggiorna a quella odierna e mi fa la richiesta all'API più recente.

```
In [ ]: datetime_object = str(date.today())
        print(datetime_object)

In [ ]: api_base_url = "https://flixpatrol.com/api/v1.2/data/?set=123&streaming=656&region=93&api=7dhd6194kdk4&date="

In [ ]: api_base_url_today = api_base_url + str(date.today())
        print(api_base_url_today)
```

Imposto il producer, che è stato impostato in locale.

```
In [ ]: producer = KafkaProducer(
            bootstrap_servers=["localhost:9092"],
            value_serializer=lambda v: json.dumps(v).encode("utf-8"))
```

Come detto in precedenza il ciclo while è stato impostato con un timer di 24 ore che per ogni ciclo fa la richiesta get all'API, il risultato viene caricato,convertito in json e mandato nella coda kafka sotto il topic "exam".

```
In [ ]: while 1:
            response = requests.get(api_base_url + str(date.today()))
            producer.send(topic="exam", value=response.json())
            time.sleep(60*60*24)
```

Fig. 2. Diagramma Flusso Apache Kafka

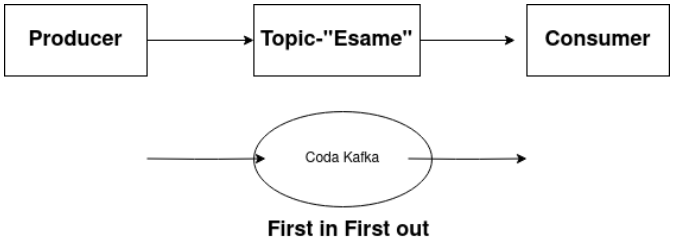
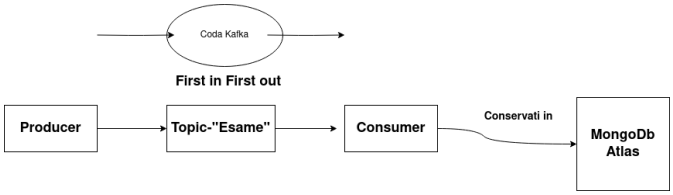


Fig. 3. Questa fase è stata sviluppata utilizzando il linguaggio di programmazione Python. Di seguito si osserva un esempio che comprende anche una sezione che verrà analizzata successivamente

```
In [ ]: from kafka import KafkaProducer
        import json
        import time
        import requests
        from pprint import pprint
        from pymongo import MongoClient
        from datetime import date

la funzione date.today() mi rida la data di oggi, che poi verrà aggiunta nel ciclo while che ha un timer di 24 ore, quindi ogni 24 ore la data si aggiorna a quella odierna e mi fa la richiesta all'API più recente.

In [ ]: datetime_object = str(date.today())
        print(datetime_object)

In [ ]: api_base_url = "https://flixpatrol.com/api/v1.2/data/?set=123&streaming=656&region=93&api=7dhd6194kdk4&date="

In [ ]: api_base_url_today = api_base_url + str(date.today())
        print(api_base_url_today)

Imposto il producer, che è stato impostato in locale.

In [ ]: producer = KafkaProducer(
            bootstrap_servers=["localhost:9092"],
            value_serializer=lambda v: json.dumps(v).encode("utf-8"))

Come detto in precedenza il ciclo while è stato impostato con un timer di 24 ore che per ogni ciclo fa la richiesta get all'API, il risultato viene caricato,convertito in json e mandato nella coda kafka sotto il topic "exam".

In [ ]: while 1:
            response = requests.get(api_base_url + str(date.today()))
            producer.send(topic="exam", value=response.json())
            time.sleep(60*60*24)
```

Fig. 4. Diagramma Flusso Apache Kafka + MongoDB Atlas



B. Memorizzazione

Il processo di memorizzazione è stato sviluppato attraverso l'utilizzo del **Cloud di MongoDB Atlas** (vedi figura 4). Considerando il formato dei dati JSON, l'idea iniziare era quella di utilizzare MongoDB per il salvataggio dei dati all'interno della macchina virtuale fornita dall'Università. L'intenzione era quella di trovare una soluzione che permettesse a tutti i componenti del gruppo di disporre dei dati in ogni istante, senza dover necessariamente interfacciarsi con la macchina virtuale per i singoli passaggi di integrazione dei dati. Tale obiettivo è stato raggiunto grazie all'utilizzo gratuito dello spazio di memorizzazione offerto da **MongoDB Atlas**. Tutto il processo è stato effettuato tramite **Pymongo**, una libreria Python utile per interagire con il database di MongoDB e che ha permesso di sfruttare MongoDB Atlas. In primo luogo è stato necessario configurare il **Cluster** per mezzo dell'interfaccia grafica presente sul sito internet di Atlas. Per la memorizzazione, utilizzando la libreria Pymongo: è stato effettuato il collegamento al **Client** di MongoDB Atlas e successivamente al **database** e in fine, per ogni mese, è stata realizzata una **collection** dove sono stati poi memorizzati tutti i dati relativi alle Top Ten.
Successivamente i dati sono stati prelevati dalla coda Kafka utilizzando il Producer configurato in precedenza, per essere poi caricati giornalmente all'interno della collection attraverso un ciclo for. Queste operazioni hanno portato alla conclusione della fase Velocità del progetto.

4. VARIETÀ : INTEGRAZIONE DEI DATI

A. Scelta del formato target del dataset

Per prima cosa abbiamo deciso che il formato finale del dataset dovesse essere quello tabellare.
Questa scelta è stata motivata da varie considerazioni: per prima cosa i dataset IMDb con cui dovevamo fare integrazione erano in formato tabellare, quindi avere un file *csv* con cui fare una semplice operazione di *join* avrebbe semplificato le operazioni per incrociare le informazioni. Inoltre era nostra intenzione arricchire i dati trovati, che finora consistevano in una sola lista dei titoli finiti in classifica nei mesi considerati, con tutte le caratteristiche di ognuno di questi titoli, come la durata per i film, la durata per episodio per le serie tv, il paese di produzione, il cast e il genere; questo perché il nostro scopo era quello di non solo calcolare gli spettacoli più popolari, ma di studiarne tutte le particolarità. Poiché bisognava arricchire ogni titolo più o meno con le stesse caratteristiche (infatti per ogni titolo ho la voce "cast", la voce "genere", "anno di produzione" eccetera), a quest'ultimo scopo si prestava bene un dataset in formato tabellare in cui ogni riga corrisponde al titolo di una serie-tv oppure un film, e nelle colonne ho le varie caratteristiche dei titoli (per esempio la colonna "cast").
Infine non è stato difficile adattare i dati che rappresentavano le classifiche nel tempo ad un formato relazionale: nel dataset finale dei titoli di Netflix ogni riga rappresenta uno show in classifica in un giorno giorno, con l'indicazione della classifica in cui si trovava, della data e della posizione nella classifica; ogni giorno è rappresentato da un insieme di 30 righe (cioè le tre classifiche quotidiane, ciascuna composta da 10 posizioni); e tutto l'insieme di righe costituisce i dati delle classifiche Netflix di ogni giorno dei quattro mesi compresi tra Dicembre e Marzo. Quello appena descritto è il formato finale del dataset in cui abbiamo salvato i nostri "dati veloci".
Oltre a questo abbiamo costruito un secondo dataset, che contiene i dati arricchiti, cioè i titoli con tutte le loro caratteristiche descritte poco fa, e senza indicazione temporale dei giorni in cui sono comparsi in classifica.

B. Da MongoDB Atlas a un dataset CSV

Questo paragrafo descrive la realizzazione del primo dei due dataset, quello dei dati relativi alle classifiche, che, acquisiti tramite MongoDB Atlas in formato JSON, abbiamo trasformato in formato *csv* usando Python.
Il processo è contenuto nel file "**Atlas**" della cartella "Variety", in formato Jupyter Notebook.
Per prima cosa grazie alla libreria **Pymongo** abbiamo ottenuto il database documentale che consisteva di tutti i dati raccolti tramite l'API di FlixPatrol, divisi in quattro collezioni che corrispondono ai quattro mesi di raccolta dati. Dopo aver estratto le quattro collezioni e averle messe in una lista, per ciascuna effettuiamo una query che trovi tutti i documenti, e li appendiamo in una lista. Otteniamo così una lista di tutti i documenti delle quattro collezioni; ogni documento rappresenta un giorno, e ha al suo interno le tre classifiche Netflix relative a quella data. Una parte del codice è mostrata in figura 2.
A questo punto, poiché i documenti MongoDB sono dizionari Python, è facile creare un dataframe usando **Pandas** che li contenga tutti. Abbiamo dunque creato il dataset dei dati Netflix con i riferimenti temporali descritto nel paragrafo precedente. Contiene 3630 righe, cioè $3630 = 31 * 30 + 31 * 30 + 28 * 30 + 31 * 30$ ossia il numero dei giorni dei quattro mesi considerati moltiplicati per le trenta righe di classifiche giornaliere.

C. Integrazione con IMDb

Come già accennato, IMDb ha messo a disposizione un dataset relazionale liberamente scaricabile dal sito, chiamato "ratings"

```
data = []

for i in range(n_collezioni):

    # Da ogni collezione estraggo con una query tutti i documenti, poi li manipolo
    cursore = collections[i].find({})
    l = list(cursore)

    for i in range(len(l)):
        del l[i]['_id']

    # Ora converto la lista dei documenti trovati in una stringa json con il metodo json.dumps
    stringa = json.dumps(l)

    # Il metodo json.loads converte la stringa serializzabile in una lista di dizionari (uno per ogn giorno del mese)
    data.append(json.loads(stringa))
```

Fig. 5. Codice delle query per estrarre tutti i documenti dalle collezioni MongoDB

e che contiene il valore medio dei voti lasciati dagli utenti su IMDblper ogni film o serie tv. Ogni titolo è identificato da una costante alfanumerica, presente anche nel database scaricato da FlixPatrol, quindi è bastato creare un nuovo dataframe estraendo i titoli dal database Netflix, e fare un'operazione di *left join* tra quest'ultimo e ratings, usando sempre la libreria Pandas. Il dataframe risultante si chiama "ratings-imdb".
Durante il processo di estrazione dei titoli ci siamo accorti di un problema minore che riguarderà tutta l'integrazione: nei dati Netflix sono presenti alcuni titoli che contenevano un apostrofo, e questo è stato reso graficamente con una stringa costante (per esempio "The Queen's Gambit" era riportato come "The Queen's Gambit").
Perciò nel file "**Integrazione imdb**" (sempre nella cartella "Variety") che contiene il codice per le operazioni descritte in questo paragrafo, è presente una funzione che corregge questi titoli, usando le espressioni regolari e la libreria Python **re**. In tutto 9 titoli su 208 presentano questa anomalia. Questa funzione è presente anche in altri script relativi all'integrazione.

D. Integrazione con Wikipedia

Abbiamo il dataset delle classifiche nel tempo, abbiamo quello dei ratings, ora dobbiamo solo creare il database dei titoli con i dati arricchiti, da incrociare infine con le votazioni IMDb.
Sempre utilizzando Pandas, abbiamo creato per prima cosa una lista con i titoli estratti dal database Netflix. Dopo aver pulito, come descritto nel paragrafo precedente, i titoli con apostrofo, abbiamo creato un nuovo dizionario che farà da scheletro al dataframe da creare, con le chiavi "Titolo", "Cast" eccetera che diventeranno le colonne. Per ottenere le informazioni abbiamo fatto **scraping da Wikipedia**, utilizzando le librerie **BeautifulSoup** e **wikipediaapi**; quest'ultima è una libreria che trova, data una stringa, l'URL della pagina Wikipedia corrispondente alla risorsa in essa descritta.
Per usarla bisogna impostare in quale Wikipedia locale svolgere la ricerca: noi abbiamo scelto Wikipedia inglese perché da una parte tutti i titoli del dataset Netflix sono scritti nella loro traduzione inglese, dall'altra essendo film internazionali, e nella maggior parte dei casi statunitensi, era ragionevole aspettarsi che il portale inglese sarebbe stato più al completo degli altri.

D.1. Disclaimer sulla lingua

I titoli in inglese corrispondono al titolo con cui lo show in questione è noto negli USA (Netflix è una società statunitense): in molti casi corrispondevano al titolo originale dello spettacolo, in altri a quello tradotto per il pubblico americano, come per esempio la serie tv spagnola di grande successo nota in Italia come "La casa di carta" (titolo originale: "La casa de papel"), memorizzata nel nostro dataset come "Money Heist", che è il titolo con cui è nota negli Stati Uniti.
tesso discorso per tutti i film e le serie tv italiane, come il film "L'incredibile storia dell'Isola delle Rose", presente nel dataset sotto il nome di "Rose Island".
I titoli italiani considerati, presenti in una certa quantità dato che i nostri dati sono le tendenze di Netflix Italia, sono spesso non molto noti in America e quindi con pagina Wikipedia inglese molto scarna; in questi casi, dopo una prima fase di scraping da Wikipedia inglese, abbiamo ripetuto lo stesso processo impostato il portale italiano; questo ovviamente solo per gli spettacoli per i quali la prima volta non eravamo riusciti a reperire abbastanza informazioni.

D.2. Fasi del lavoro

Il codice descritto qui è contenuto nel file "**Integrazione wikipedia**".
Il lavoro di integrazione è stato diviso nei seguenti passi:

1. Creazione della funzione **wikipages**: funzione che riceve in ingresso il database Netflix e l'elenco dei titoli di tutti gli show sotto forma di lista, e restituisce un dizionario in cui le chiavi sono i titoli della lista, e, il valore corrispondente in ogni chiave è la pagina Wikipedia inglese che descrive il fil o la serie tv con quel titolo. La pagina è l'oggetto *page* della libreria wikipediaapi, che contiene, tra le altre cose, l'URL della risorsa in Wikipedia. La funzione cerca tramite wikipediaapi, che ha un'apposita funzione di ricerca tramite parola chiave, la pagina corrispondente al titolo, nel modo più accurato possibile. In particolare, la funzione usa il database dato in ingresso per controllare il tipo di un titolo (ossia se esso è una serie tv oppure un film) e usa questa informazione per cercare in modo più accurato la pagina. Quasi sempre infatti, non basta cercare solo il titolo dello show che si vuole per ottenere la pagina corrispondente, ma spesso bisogna aggiungere la stringa canonica in Wikipedia "(TV series)" oppure "(Movie)".
   Ci sono molti casi particolari, per esempio serie tv che hanno lo stesso nome di film, oppure film con lo stesso nome di altri film usciti in anni diversi eccetera. In sintesi: se la funzione trova la pagina p corrispondente al titolo t, allora nel dizionario risultante alla chiave t corrisponderà la pagina p; se invece la funzione non riesce a trovare nessuna pagina per t, allora alla chiave t corrisponderà la stringa "0". Il dizionario restituito è stato chiamato **diz**.

2. Poiché per tre titoli wikipages effettivamente trova una pagina Wikipedia, ma si tratta della pagina di desambiguazione, abbiamo corretto questi tre elementi del dizionario mettendo come chiave, a mano, la stringa "0". Successivamente abbiamo contato che i titoli con valore "0" sono 18, perciò per 18 titoli su 208 non abbiamo trovato in modo automatico una pagina Wikipedia corrispondente da cui reperire informazioni.

3. Prima fase di **scraping**: utilizzando la libreria Python **BeautifulSoup**, abbiamo ricavato, per ogni titolo con una pagina del dizionario (quindi con valore corrispondente diverse da "0"), un dataframe dall'infobox laterale presente in ogni pagina Wikipedia. La forma standardizzata di questi infobox ha reso più semplice il processo.
   In particolare, è stato creato un nuovo dizionario, **diz_ tab**, in cui ad ogni titolo corrisponde il dataframe con le informazioni dell'infobox di Wikipedia inglese (in realtà solo per i titolo per cui è stata trovata una pagina dalla funzione

wikipages, perciò tutti meno i 18 titoli contenuti nella lista ausiliare apposita **titoli_mancanti**, creata per tenere traccia degli spettacoli per cui non eravamo ancora riusciti a reperire informazioni). In questa fase anche i titoli senza infobox vengono aggiunti a titoli_ mancanti.
Il codice è contenuto nella **cella 22** del file.

4. Successivamente devo manipolare i dataframe ricavati dagli infobox del punto precedente e contenuti in diz_tab. In particolare, lo scopo era costruire due database, uno per le serie tv e uno per i film, in cui ad ogni riga corrisponde uno spettacolo, e le colonne ne descrivono le caratteristiche ricavate dagli infobox. I database sono divisi perché gli infobox delle serie tv e dei film su Wikipedia sono leggermente diversi (per fare un esempio, una differenza è che per le serie tv si trova "created by", invece per i film "written by"), perciò era più comodo dividerli in questa fase e poi riunirli nell'integrazione finale.
A questo scopo servono le funzioni **crea_df_film** e **crea_df_serie** (celle 28-29), esse vengono eseguite nella cella 30, e nelle celle successive sono stati creati i due dataframe.

5. Non eravamo solo interessati alle informazioni contenute nell'infobox, che pur conteneva già parecchi dati, ma anche a quelle contenute in alcune sezioni delle pagine Wikipedia, facilmente accessibili tramite il metodo *section_by_title* dell'oggetto *page* di wikipediaapi.
Perciò l'idea è stata quella di creare delle nuove colonne per ciascuno dei due dataframe (codice contenuto nelle **celle 34-39**), in base alle sezioni standardizzate di Wikipedia, ancora una volta diverse se si tratta di una pagina che descrive un film oppure una serie tv.
In particolare per le serie tv abbiamo aggiunto al dataframe le colonne: "Sommario" (corrispondente al sommario nelle pagine Wikipedia), "Cast_testo" (intera sezione del cast in formato testuale), "Produzione_testo" (sezione della produzione), "Critica" (ricezione da parte della critica).
Invece per i film le colonne aggiunte sono state: "Sommario" (sommario della pagina, generalmente non si limita solo alla trama del film, ma ne descrive altre caratteristiche come il soggetto o la produzione), "Trama" (trama del film), "Cast_testo" (come prima, sezione del cast in formato testuale), "Produzione_testo" (sezione della produzione), "Critica" (ricezione da parte della critica).

6. A questo punto ci occupiamo dei titoli mancanti: i dataframe creati contengono infatti solo i titoli per cui è stata trovata sia la pagina Wikipedia sia l'infobox. Per la precisione abbiamo avuto solo un caso, "My Hero Academia", in cui la pagina era stata trovata correttamente ma non l'infobox, e questo era dovuto alla particolare strutture delle varie voci Wikipedia riferite a questo titolo.
In generale tutti i titoli per cui finora non eravamo riusciti a ricavare dati, contenuti nella lista **titoli_mancanti** che a questo punto ammontava a 26 elementi, dovevano essere trattati.
Il codice qui descritto è contenuto nella sezione "**Gestione titoli mancanti**".
In analogia con la prima parte, abbiamo creato un dizionario, **diz_mancanti**, che, per ognuno di questi titoli, contiene il tipo dello show (cioè se è una serie tv oppure un film) e l'URL della pagina Wikipedia a cui reperire le informazioni, se esso è stato trovato con facilità.
Abbiamo provato inizialmente ad aggiungere questi URL in modo automatico, utilizzando wikipediaapi impostando la lingua italiana, tuttavia l'esperimento si è rivelato fallimentare, dato che tra i 26 titoli mancanti, solo 10 avevano come paese d'origine l'Italia, e pur usando la libreria per cercare le pagine di questi, è stata trovata automaticamente solo una pagina su dieci. Quindi abbiamo preferito, visto il numero esiguo di titoli da trattare, inserire l'URL a mano.
Una volta inseriti gli indirizzi delle pagine Wikipedia su cui fare scraping, abbiamo creato due nuovi dataframe, chiamati **Mancanti_serie** e **Mancanti_film**, provvisti delle stesse colonne rispettivamente del dataframe delle serie-tv e di quello dei film; l'idea era concatenare questi due nuovi dataframe con le informazioni di titoli mancanti a quelli principali.
Per ogni elemento di diz_mancanti siamo dunque andati avanti con lo scraping in modo molto simile a quanto avevamo fatto con il primo gruppo di titoli, utilizzando ancora la libreria BeautifulSoup, catturando prima le informazioni contenute nell'infobox e trasportandole nel dataframe, poi quelle testuali contenute nella pagina, salvate in delle liste ausiliarie, poi trasformate in nuove colonne dello stesso.
Infine abbiamo concatenato questi ultimi dataframe a quelli principali, ottenendo così due dataframe completi, uno per le serie-tv e uno per film. In essi sono presenti complessivamente tutti i 208 titoli, tra di essi ci sono stati alcuni titoli per cui ancora non eravamo riusciti ad estrarre automaticamente alcuna informazione. Il loro numero (16 titoli) era esiguo rispetto al totale degli show studiati, perciò abbiamo deciso di inserire i dati a mano, anche perché considerando la loro eterogeneità era difficile pensare a uno schema per estrarre informazioni su tutti loro contemporaneamente, o comunque avrebbe richiesto più tempo piuttosto che inserire a mano poche informazioni.
I risultati di questa fase sono i dataframe **film_wiki** e **serie_tv_wiki**.

E. Integrazione finale

L'ultimo script Python relativo all'integrazione è contenuto nel file "**Integrazione finale**".
Lo scopo di questo file è manipolare i dataframe finora creati e trasformarli in due dataset definitivi, uno per i film e uno per le serie-tv, in cui per ogni show sia inoltre calcolato l'indice di popolarità. Per prima cosa abbiamo importato tutti i dataframe, e abbiamo incrociato tramite la funzione **merge** di Pandas i dataframe di Wikipedia con quelli di IMDb, e i dataset risultanti sono **movies** e **tv_series**.
Successivamente abbiamo preparato i dati per il calcolo dell'indice di popolarità: ciò richiedeva di cercare nel dataset Netflix ogni titolo, tuttavia i titoli che presentavano un apostrofo avevano solo in questo dataframe una grafia particolare; perciò abbiamo compiuto l'ormai usuale operazione di estrarre questi titoli, correggerli e salvarli in una lista a parte, per agevolare le operazione di scorrimento nel dataset. Il calcolo richiedeva infatti di scorrere tutte le classifiche, al fine di computare i giorni in cui ognuno dei titoli era stato in classifica, oltre alla classifica in cui era stato e la posizione; ed è stato implementato mediante tre funzioni diverse dalla **cella 16** alla **cella 18**. Queste tre funzioni sono: **indice_popolarità_gen**, **indice_popolarità_film**

e **indice_popolarità_serie**, e, come suggeriscono i nomi, calcolano l'indice rispetto alla classifica generale, dei soli film e delle sole serie-tv. Esse effettuano operazioni riga per riga sui dataframe movies e tv_series: per ogni riga estraggono il titolo dello show, e utilizzando il dataset Netflix calcolano il suo indice di popolarità. Sono applicate ai due dataframe nella **celle 19** e **21** utilizzando la funzione **apply**.
Osserviamo che uno dei motivi per cui è stato conveniente, almeno fino a questo momento, tenere i due dataset separati è proprio il calcolo dell'indice: in questa fase abbiamo potuto computare separatamente l'indice tipo delle serie tv e dei film, servendoci dei dataframe ausiliari appositamente creati *top_film* e *top_serie* (per la definizione di indice tipo rimandiamo al paragrafo 2).
Per entrambi i dataset, l'indice generale e l'indice tipo sono stati salvati nelle rispettive colonne. Abbiamo scelto di chiamare la colonna di entrambi "Indice_tipo" per rendere più facile l'operazione di concatenazione successiva.
Infine, nell'ultima parte dello script abbiamo calcolato i cinque spettacoli con indice di popolarità più alto.
A tal fine i due dataset sono stati concatenati, è stata creata la colonna "Popolarità" come somma delle due colonne che rappresentano indici, e infine le righe appartenenti al dataframe risultante sono state ordinate in modo discendente rispetto alla colonna appena creata. I cinque titoli più popolari (tutte serie-tv) sono risultati:

1. **Bridgerton**, serie statunitense in costume di genere sentimentale, comparsa su Netflix il giorno di Natale del 2020;
2. **The Queen's Gambit**, miniserie nota in Italia come "La regina degli scacchi", uscita ad Ottobre del 2020 e anch'essa proveniente dagli Stati Uniti;
3. **Lupin**, serie francese a metà tra il poliziesco e il genere d'azione, la sua premierè su Netflix è stata a Gennaio del 2021;
4. **Ginny & Georgia**, serie statunitense collocata tra la commedia e il genere drammatico, uscita a fine Febbraio del 2021;
5. **New Amsterdam**, serie statunitense di genere *medical drama*, e l'unica della Top 5 a non avere Netflix come casa di produzione.

5. PULIZIA DEL DATASET

Una volta completata la fase di integrazione tra i dataset, abbiamo effettuato una fase di pulizia dei dati.
La pulizia dei dataset è stata operata in parte manualmente e in parte tramite script in Python. In particolare gli attributi maggiormente interessati dalla pulizia sono stati:

1. **Paese**: la colonna presentava righe in cui comparivano più nazioni; tale difetto avrebbe potuto produrre problemi di analisi successive. Attraverso l'utilizzo di espressioni regolari implementata da Excel, dove vi erano caratteristiche di formattazione uguali, è stato applicata la correzione.
   Diversamente in altri casi è stato necessario procedere con una modifica manuale;
2. **Produzione**: la colonna presentava righe in cui comparivano più case di produzione; non trovando casi di formattazione riconoscibile in un pattern, è stato necessario procedere con una modifica manuale;
3. **Distribuzione**: come per le case di produzione, questa colonna presentava righe in cui comparivano più case di distribuzione; non trovando pattern riconoscibili, è stato necessario procedere con una modifica manuale;
4. **Durata**: in alcune righe la durata era rappresentata da un range, perciò è stato necessario utilizzare una funzione scritta in Python che, sfruttando una ricerca di pattern, ha permesso di calcolare la media; negli altri casi è stato necessario procedere con la correzione manuale.

Inoltre è stato realizzato un raggruppamento dell'attributo "Genere"; quest'operazione di *preprocessing* è risultata necessaria perché questo attributo presentava un numero molto elevato di modalità, rendendo complessa l'estrazione di informazioni di valore. Abbiamo sintetizzato questo attributo nel nuovo "Genere_unico", che contiene, per ogni titolo, il genere più rappresentativo, scelto fra i diversi generi indicati per quel titolo. Per aiutarci in questa operazione abbiamo talvolta consultato anche il sommario o la trama dello show in questione.
Abbiamo ottenuto 15 generi raggruppati:

1. **Action** : genere d'azione;
2. **Adventure** : genere d'avventura;
3. **Animation** : film o serie-tv di animazione, comprende dagli *anime* giapponesi ai cartoni per bambini;
4. **Biography** : biografici, generalmente film;
5. **Comedy** : genere comico, comprende dai film comici alle *sit-com*;
6. **Crime** : poliziesco o giallo;
7. **Documentary** : documentari, comprende sia film sia serie-tv;
8. **Drama** : genere drammatico;
9. **Dramedy** : genere a metà tra "comedy" e "drama";
10. **Family** : spettacoli per tutta la famiglia;
11. **Fantasy** : genere fantastico;
12. **Horror** : genere horror;
13. **Romance** : genere sentimentale;
14. **Science fiction** : fantascienza;
15. **Thriller**.

6. ANALISI ESPLORATIVA DEI DATASET FINALI

Il nostro lavoro ha prodotto due dataset: il primo, quello scaricato dall'API di Flixpatrol.com, contiene i dati relativi alle classifiche di Netflix ed è stato usato per calcolare l'indice di popolarità di tutte le serie-tv e i film; il secondo è invece il dataset che contiene i dati arricchiti per tutti i film e le serie tv.
Abbiamo ottenuto complessivamente 208 titoli, di cui 139 film e 69 serie-tv, come mostrato in figura 6.

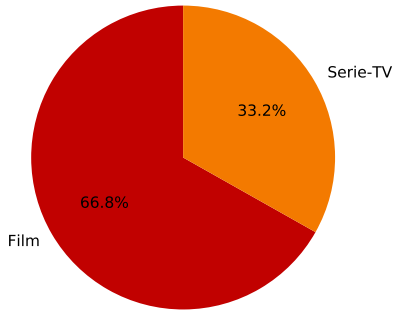


Fig. 6. Nel dataset abbiamo 139 film e 69 serie-tv

A. Osservazioni generali sulla popolarità dei titoli

L'indice di popolarità, che come è specificato nel paragrafo 2, è compreso tra 0 e 2, ha in generale un valore molto basso, basti pensare che la sua media nel dataset è 0.079, mentre la mediana è di 0.039.
Come è evidente dalla figura 7, la maggior parte degli show presenta un indice di popolarità molto inferiore a 0.2, e ovviamente nessuno show ha indice esattamente pari a zero.
Gli elementi riassuntivi della distribuzione sono:

- **Valore minimo** = 0.000826
- **Primo quartile** = 0.0099
- **Mediana** = 0.0388
- **Terzo quartile** = 0.0975
- **Valore massimo** = 0.8636

La maggior parte degli spettacoli hanno ovviamente indice compreso tra 0.01 e 0.039, ma dalla figura 7 è evidente che esista un nutrito gruppo di outlier, che rappresentano le **tendenze** su Netflix in questi mesi, ossia gli show che semplicemente arrivano ad essere di moda. La dinamica delle tendenze su Netflix in questi mesi è quindi costituita da un grande varietà di spettacoli, dei quali molti sono entrati in classifica ma ci sono rimasti per poco tempo, e invece pochi si sono affermati come tendenza e sono riusciti a rimanere in classifica per molti giorni. In tutto sono 52 i film e le serie-tv con indice di popolarità superiore a 0.0975 (il terzo quartile), tra i quali solo 18 hanno indice superiore a 0.2. Sono poi evidenti dal boxplot gli outlier che si riferiscono ai primi tre show, che commenteremo in modo approfondito nel paragrafo finale.

B. Varietà di generi

Come mostrato in figura 8, i generi a cui appartengono più spettacoli sono *Action* e *Drama*, entrambi con 37 titoli, il che non ci ha sorpreso dato che sono entrambe categorie molto generiche, in cui si ritrovano trame anche molto differenti tra loro; per esempio il genere *Drama* comprende dal capolavoro di Clint Eastwood "Million Dollar Baby" alla serie-tv di genere *legal-drama* "Suits", incentrata su uno studio legale.
Oltre a questi e al genere *Comedy*, che conta 31 titoli, gli altri *bin* rappresentano specie di film molto più specifiche, e di

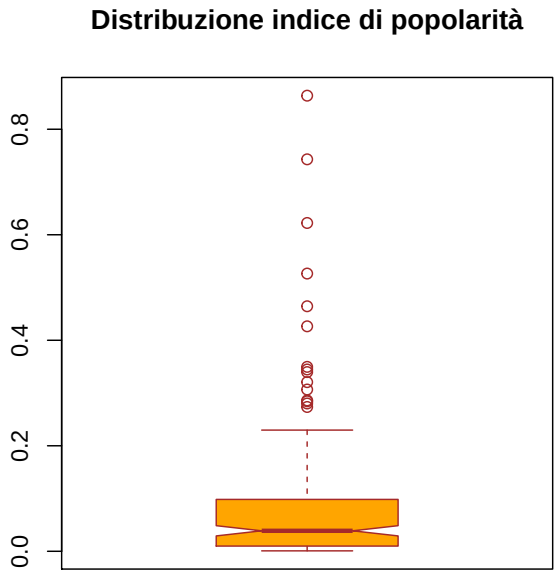


Fig. 7. Popolarità degli show su Netflix tra Dicembre 2020 e Marzo 2021

conseguenza anche il numero di titoli che ad essi appartengono è esiguo: è il caso di *Documentary*, *Horror*, e *Animation*.

C. Altre caratteristiche

Possiamo notare come Netflix sia la casa di distribuzione per 86 titoli su 208 (il 41,35%), considerando solo i titoli che distribuisce direttamente e non insieme ad altre reti locali o internazionali. Ciò non stupisce, come nemmeno il fatto che il primo paese di provenienza dei titoli siano gli Stati Uniti, patria dell'azienda, sono infatti 122 tra film e serie-tv ad avere come paese di provenienza solamente gli USA (senza considerare le co-produzioni), più del 58% del totale.

D. Qualità secondo IMDb

Come già accennato, abbiamo misurato i nostri titoli mediante i voti lasciati dagli utenti di IMDb; in particolare per ogni film, o serie-tv, abbiamo chiamato "**qualità**" del titolo la **media** dei voti degli utenti IMDb per quel film o quella serie tv.
La "qualità" è intesa come valore dello spettacolo in questione in quanto opera artistica nel suo complesso: esso può essere pregevole per molte ragioni, come la sua trama o l'originalità del soggetto, o ancora grazie alle performance degli attori che vi hanno recitato. Gli utenti riassumono l'impressione complessiva che uno spettacolo televisivo ha lasciato loro nel *rating* IMDb, compreso tra un **minimo di 1** e un **massimo di 10**.
La figura 6 mostra la distribuzione della qualità dei nostri titoli in classifica, e si nota subito come la mediana sia piuttosto alta: essa è pari a 6.7. La media le è molto vicina: è 6.649. Ciò indica che le serie-tv e i film più visti dagli utenti Netflix sono anche prodotti di qualità, che quindi diventa un fattore importante nel

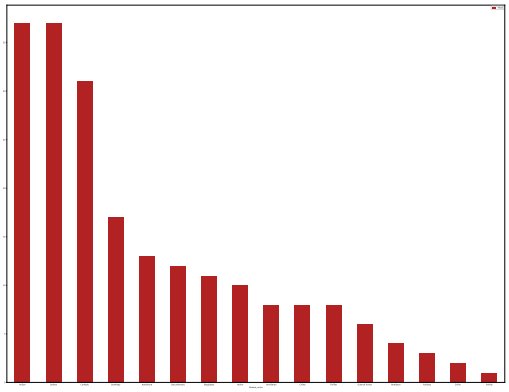


Fig. 8. Conteggio per genere

determinare un successo di un titolo.
Gli altri indici statistici sono:

- **Valore minimo** : 3.2
- **Primo quartile** : 5.875
- **Mediana** : 6.7
- **Terzo quartile** : 7.4
- **Valore massimo** : 8.7

L'essere un prodotto di pregio non è caratteristica di ogni titolo presente nelle classifiche Netflix di questi mesi: il film polacco "365 days" che realizza il minimo valore di qualità del dataset (3.2) è un caso di successo sulla piattaforma nonostante le numerose critiche negative (fonte: 365 giorni).
Il valore massimo (8.7) è ottenuto da due serie-tv molto diverse tra di loro: la prima è "Formula 1: Drive to Survive", serie di genere *Documentary* che racconta il mondo della Formula 1; la seconda è la drammatica "The Crown", che narra in modo romanzato la vita della sovrana inglese Elisabetta II; a conferma della qualità di quest'ultima serie, basta dire che sia stata acclamata dalla critica e che abbia vinto numerosi premi, come specificato nella colonna "Sommario". Sono entrambe inglesi e prodotte da Netflix.

7. ANALISI DEI RISULTATI

L'analisi dei risultati è rivolta ad individuare quali sono le caratteristiche degli show che hanno riscosso maggiore successo tra il pubblico italiano.

A. Popolarità: film vs serie-tv

In figura 7 sono riportate le distribuzioni, di cui è segnato anche il *notch*, dell'indice di popolarità nei soli film e nelle sole serie-tv; è evidente come la distribuzione della popolarità delle serie-tv sia spostata verso valori più elevati rispetto ai film, anche senza considerare i numerosi outlier della categoria, che risulteranno essere proprio gli show più visti di Netflix.
In particolare gli indici statistici sono:

- **Minimo**: 0.00083 (film), 0.00083 (serie-tv);

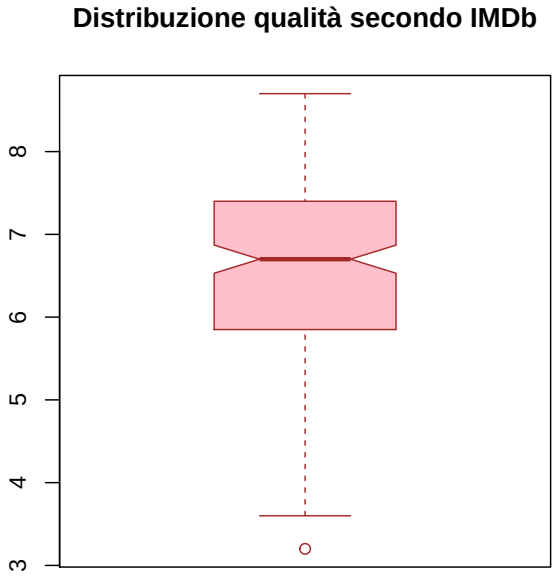


Fig. 9. Distribuzione della qualità media degli show Netflix

- **Primo quartile**: 0.0099 (film), 0.019 (serie-tv);
- **Mediana**: 0.028 (film), 0.062 (serie-tv);
- **Terzo quartile**: 0.076 (film), 0.146 (serie-tv);
- **Massimo**: 0.321 (film), 0.864 (serie-tv);
- **Media**: 0.052 (film), 0.135 (serie-tv).

Il minimo è lo stesso, ma la mediana delle serie-tv è molto più alta (di più di 3 centesimi), così come i quartili e la media. In questo caso sono presenti molti outlier per entrambe le distribuzioni, e quindi la media ne è modo influenzata e risulta molto maggiore della mediana in entrambi i casi.
Si conclude che il tipo di show preferito dal pubblico italiano su Netflix sono sicuramente le **serie-tv**.
Poniamo lo sguardo sui numerosi outlier, ossia gli show che si sono distinti, tra i film o le serie-tv, perché hanno avuto un successo molto maggiore degli altri.

A.1. Oulier per i film

Gli outlier sono gli spettacoli per cui il valore dell'indice di popolarità è superiore al terzo quartile più 1.5 moltiplicato per la differenza interquartile (IQR) oppure quelli che hanno indice minore del primo quartile meno 1.5 moltiplicato per IQR.
Gli outlier che rientrano nel primo caso hanno una popolarità notevole in positivo, cioè si distinguono dalla massa perché hanno avuto più successo, quelli del secondo tipo invece si distinguono perché ne hanno avuto molto meno rispetto alla generalità degli altri spettacoli del dataset.
Tra i film non sono presenti outlier del secondo tipo, cioè particolarmente poco popolari rispetto agli altri, ma esistono **5 film** che si sono distinti per il loro successo: andando in ordine di popolarità crescente, abbiamo "A California Christmas",

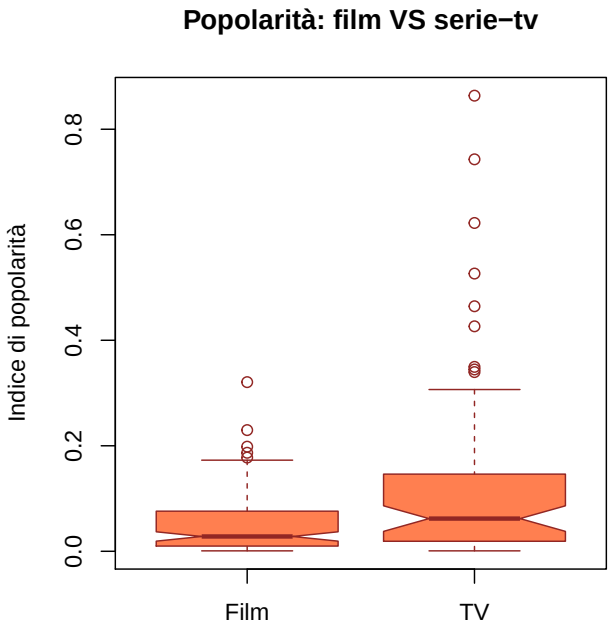


Fig. 10. Distribuzione dell'indice di popolarità suddiviso per genere degli spettacoli.

"Ava" e "A Star is Born", che hanno quasi la stessa popolarità, poi "The Christmas Chronicles: Part Two" e infine distacca tutti "Rose Island", film italiano il cui titolo originale è "**L'incredibile storia dell'isola delle Rose**".
Soprattutto il successo di questi ultimi due titoli è una controtendenza al *trend* delle serie-tv. Mentre il successo del primo film, "**The Christmas Chronicles: Part Two**", si spiega osservando che esso rientra nel genere dei "*film di Natale*" americani (infatti è uscito il 25 Novembre del 2020, distribuito da Netflix); quello de '"L'incredibile storia dell'isola delle rose" è inaspettato, sicuramente per un titolo di madrelingua italiana.
Il film ha inoltre tutti attori italiani (i protagonisti sono Elio Germano e Matilda De Angelis) e regista italiano (Sydney Sibilia); il suo punteggio di qualità secondo gli utenti di IMDb è abbastanza alto (7), ed è anche stato recensito favorevolmente dalla critica internazionale; l'essere un prodotto di qualità alta è stato probabilmente determinante per il suo successo.

A.2. Outlier per le serie-tv

Anche nel caso delle serie-tv non abbiamo outlier in negativo, cioè con popolarità particolarmente bassa rispetto alle altre, risultano invece 9 outlier positivi, che sono, in ordine crescente di popolarità : "Snowpiercer", "Fate: The Winx Saga", "Firefly Lane", "Riverdale", "New Amsterdam", "Ginny & Georgia", "Lupin", "The Queen's Gambit", "Bridgerton".
Tra questi, i primi tre titoli risultano gli show più seguiti su Netflix in assoluto, e verranno approfonditi nel paragrafo seguente.

Fig. 12. Lo show più visto di Netflix tra dicembre e Marzo: Bridgerton. Il titolo è il nome della famiglia protagonista della vicenda

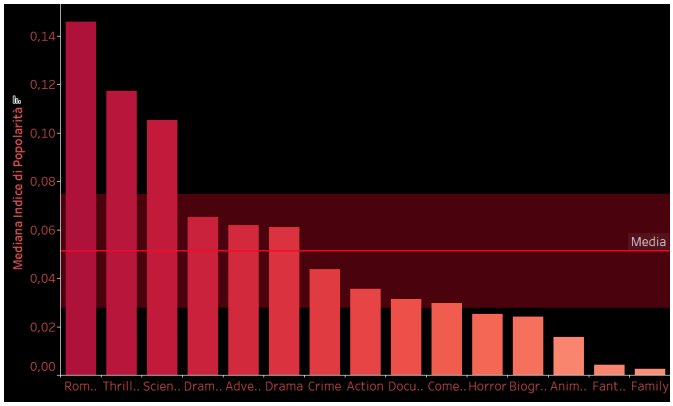


Fig. 11. Genere più popolare

B. Conclusione e sviluppi futuri

Dalle analisi condotte emerge come le serie-tv abbiano una popolarità superiore ai film, nonostante, come possiamo vedere dalla Figura 3, il numero delle serie-tv sia nettamente inferiore al numero dei film; possiamo dare molteplici spiegazioni a questa situazione, in particolare il fattore che potrebbe giocare a vantaggio delle serie-tv è il fatto che esse in questo periodo sono in voga tra i più giovani.
Abbiamo inoltre condotto un analisi più approfondita per capire qual è il genere che ha raggiunto popolarità maggiore: dalla Figura 11 vediamo che i generi che hanno Indice di popolarità più alto sono: "Romance" e "Thriller". Al contrario, i generi meno popolari sono "Family" e "Fantasy". Nella figura facciamo riferimento alla mediana in quanto facendo le analisi con la media vi era una grande influenza degli outlier, ed in particolare l'outlier riferito allo show "Bridgerton", classificato sotto il genere "Romance".

B.1. Top 3

In questo paragrafo vogliamo analizzare le caratteristiche principali dei tre show più seguiti su Netflix, che tra Dicembre 2020 e

Fig. 13. "The Queen's Gambit", in italiano *il gambetto di donna*, è un'apertura di una partita a scacchi

Fig. 14. Il protagonista Assane Diop si ispira ai romanzi con protagonista Arsenio Lupin, il ladro gentiluomo

Marzo 2021 sono stati:

1. **Bridgerton**: indice di popolarità = 0.864.
   Prodotto dalla creatrice di *Grey's Anatomy* Shonda Rhimes e distribuito da Netflix, la prima stagione di questo show è uscita il 25 Dicembre 2020 e in poco tempo si è confermata come il più grande successo tra le serie-tv originali Netflix. Il debutto nel giorno di Natale mostra come le aspettative sul successo del prodotto fossero alte. I generi a cui appartiene sono quelli di "dramma in costume", "sentimentale","drammatico", noi lo abbiamo collocato sotto il genere raggruppato *Romance*. L'ambientazione è il periodo dell'ottocentesca *Regency era* inglese, e il soggetto non è originale ma è basato su una serie di romanzi. La serie ha ottenuto una valutazione di qualità medio alta su IMDb pari a 7.7.

2. **The Queen's Gambit**, indice di popolarità = 0.743.
   Questa serie americana è nota in Italia come "La regina degli scacchi", ed è anch'essa tratta da un romanzo, il quale narra la storia immaginaria di una bambina prodigio nell'ambiente scacchistico degli anni '60. La trama prende chiaramente ispirazione dalla vera storia del grande campione di scacchi americano Bobby Fisher, soprattutto negli episodi in cui si fa riferimento alla guerra fredda. Oltre al gioco, l'altro grande tema è la dipendenza da alcool e droghe contro cui la protagonista deve lottare.
   La qualità media secondo gli utenti di IMDb è tra le più alte di tutti gli spettacoli nel dataset, raggiungendo un valore di 8.6; e in effetti la serie è stata lodata dalla critica sia per la performance dell'attrice protagonista sia per l'originalità e la riuscita della trama.

3. **Lupin**, indice di popolarità = 0.622.
   È l'unico titolo tra i primi cinque a non provenire dagli USA: proviene infatti dalla Francia, e mostra una forte identità francese, dato che il protagonista è interpretato da uno degli attori francesi più noti in tutto il mondo (Omar Sy), è ambientata a Parigi, e la trama è liberamente ispirata ai romanzi di Maurice Leblanc, padre del personaggio del ladro Arsenio Lupin. Anche questa serie è una serie originale Netflix. Il giudizio di qualità di IMDb è medio alto: è pari a 7.5. La critica ha lodato la performance di Omar Sy come attore, oltre al buon ritmo e alla sapientemente costruita *suspence* della trama, caratteristica fondamentale per una serie-tv come questa, a cavallo tra i generi "thriller", "giallo" e "drammatico". Nel progetto è stata classificata sotto il genere unico "Crime".

Tutte e tre sono serie originali Netflix. In comune, oltre alla casa di distribuzione, hanno il fatto di avere pochin episodi: "Bridgerton" conta 8 episodi, "The Queen's Gambit" è una miniserie composta da 7 puntate, e anche "Lupin" ha 10 episodi. Si può dire che nonostante siano pochi, gli episodi sono molto condensati, perché gli episodi di tutte e tre le serie hanno una lunghezza notevole, che varia dai 45 ai 70 minuti.
Inoltre tutte e tre hanno ricevuto recensioni favorevoli dalla critica, e hanno un gradimento IMDb che spazia da alto (Bridgerton) a molto alto (The Queen's Gambit), ciò è indice del fatto che per creare una serie-tv di successo è importante considerarne la qualità, in particolare le performance degli attori, e la robustezza della trama.

B.2. Sviluppi Futuri

All'interno del Dataset è contenuta una colonna (**tag_name**) che fa riferimento alle recensioni provenienti per la maggior parte da Rotten Tomatoes. L'idea che è stata sviluppata riguardava l'utilizzo di un sistema di Text Analysis, in particolare sentiment analyisis, che sfrutta l'API messa a disposizione da **MonkeyLearn**, liberamente utilizzabile.
Attraverso l'analisi dei **testi** delle recensioni sono stati restituiti alcuni tag ("positive", "negative" oppure "neutral") e le loro confidenze, che hanno permesso di classificarli come recensioni positive, neutre oppure negative. Il tutto avrebbe permesso analisi sicuramente più approfondite delle recensioni professionali da parte della critica per ogni show.
All'interno del progetto questi dati tuttavia non sono stati effettivamente utilizzati, poiché la nostra conoscenza in materia era minima e di conseguenza non eravamo certi se il metodo

selezionato potesse essere valido e di conseguenza abbiamo deciso di non perseguire questa strada, poiché l'aggiunta di dati di cui non conoscevamo l'affidabilità avrebbe potuto intaccare la qualità finale del progetto. Sicuramente il prossimo anno dopo aver affrontato il corso di Text Mining and Search potremo avere maggiore consapevolezza sui metodi adottabili in questi contesti e poter sfruttare al meglio anche queste tipologie di dati testuali.